UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA
DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

TESIS DOCTORAL

System Level Management of Hybrid Memory Systems

Gestión de jerarquías de memoria híbridas a nivel de sistema

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Manu Perumkunnil Komalan

DIRECTORES

José Ignacio Gómez Pérez
Christian Tomás Tenllado
Francky Catthoor

Madrid, 2018

KU LEUVEN

Universidad Complutense de Madrid
Facultad de Informática
Departamento de Arquitectura de Computadores y Automática

System Level Management of Hybrid Memory Systems

Gestión de jerarquías de memoria híbridas a nivel de sistema

Manu Perumkunnil Komalan

Supervisors
Prof. dr. ir. José Ignacio Gómez Pérez (UCM)
Prof. dr. ir. Christian Tomás Tenllado (UCM)
Prof. dr. ir. Francky Catthoor (KU Leuven)

March 2017

System Level Management of Hybrid Memory Systems

Gestión de jerarquías de memoria híbridas a nivel de sistema

Manu Perumkunnil KOMALAN

Examination committee:

Prof. dr. ir. José Ignacio Gómez Pérez
Prof. dr. ir. Christian Tomás Tenllado
Prof. dr. ir. Francky Catthoor
Prof. dr. ir. Wim Dehaene
Prof. dr. ir. Dirk Wouters
Prof. dr. ir. Manuel Prieto Matías
Prof. dr. ir. Luis Piñuel
Prof. dr. ir. José Manuel Colmenar

March 2017

Acknowledgments

There is an impossibly long list of people I want to thank for helping me in the pursuit of my PhD and making it worth much more than simple technical jargon. Like I've been counseled many a times and experienced, my PhD like every other PhD has followed the trajectory of a sine wave. There have been crests and there have been troughs. At the end, I'm glad the positive aspects won out and the journey provided me with the experience of a lifetime.

I want to start off by thanking my teachers who taught and guided me at different stages of my academic and work life before I started my PhD. It was their vision and guidance that finally culminated in this. A few names that come to mind are Mr. Robin Kumar, Dr. O.P. Sinha, Dr. Susmita Mitra, Dr. Sandip Chakraborty, Dr. K. S. R Koteswara Rao, Dr. R. P. Singh, Dr. Subhasis Ghosh, Dr. Shanu Chacko, Dr. John Joseph, Dr. Jaya Lal Ji and Dr. Nancy George.

I have been blessed to have a family who has always supported me, loved me unconditionally, inspired me and been the bedrock on which my life is based. My parents have provided me with strength, whenever it was lacking and it is only due to their upbringing, constant care and steadfast confidence in me that I've been able to reach where I have. Through the toughest times of my PhD they have braced me without questions. My younger brother has been another source of support and love for me who has kept me grounded and true to my self. His cheerful nature and solutions to stressful situations have always picked me up.

They say that each friendship offers something totally unique and irreplaceable. Each friendship ultimately makes us who we are. I've been lucky enough to have a plethora of friends who have helped me when I was constantly shuffling between Madrid and Leuven and also helped me whenever I was down by cheering me up and offering consolation whenever required. First and foremost, I'd like to thank my close friends in Leueven without whom, it would have been

impossible to get through the PhD. Arul, Manisha, Phalguni, Diana, Sajin, Ameya, Ashwini and Sriram... without you guys it wouldn't have been worth it. I'd like to thank Ricardo and Jorge Val Calvo for being the very best roommates during my stay in Madrid. They helped me out with the bureaucracy whenever it came up and were always up for a drink to lift my spirits. I'd like to thank all the lab mates in Madrid: Jorge Quintas, Daniel Tabas, Jose Luis, Fermin and Javier for making my stay at UCM thoroughly enjoyable. I'd also like to thank my friends and colleagues at imec, namely, Matthias, Prashant, Arindam, Trong, Bharani for making my stay at imec a wonderful experience. I'd like to especially thank Sushil Sakhare and Oh HyungRock for being more than just friends and also mentoring me in whatever way possible.

I would also like to thank the numerous mentors who have helped me over the course of my PhD like Dr. Dirk Wouters, Dr. Stefan Cosemans, Dr. Julien Ryckaert, Dr. Praveen Raghavan, Prof. Manuel Prieto and Prof. Jose Francisco Tirado. I would also like to thank Prof. Wim Dehaene for taking the time to review my PhD thesis and being an invaluable part of my supervisory committee. Finally, I would like to thank the three most important people who have been responsible for me completing this PhD: Prof. Francky Catthoor, Prof. José Ignacio Gómez and Prof. Christian Tenllado. Francky has mentored me not only technically but also on various aspects of life and broadened my horizons. He has always had a solution to any problem and with unending energy helped me whenever possible and in whatever way. I will always be grateful for that. I cannot stress how much of a help Nacho (José Ignacio) has been to me these last 5-6 years. He has been more of an elder brother to me than an advisor and helped me in innumerable ways, be it technically, financially or any other. It is only due to his help that I was able to navigate through one of the darkest phases of my PhD. Christian has been the same and helped me immensely whenever possible.

Besides work, living in Europe has also been a fascinating experience. After about 6 years, I feel like I can truly call Leuven my second home.

Contents

List of Figures

List of Tables

Abstract

In electronics and computer science, the term 'memory' generally refers to devices that are used to store information that we use in various appliances ranging from our PCs to all hand-held devices, smart appliances etc. Primary/main memory is used for storage systems that function at a high-speed (i.e. RAM). The primary memory is often associated with addressable semiconductor memory, i.e. integrated circuits consisting of silicon-based transistors, used for example as primary memory but also other purposes in computers and other digital electronic devices. The secondary/auxiliary memory, in comparison provides program and data storage that is slower to access but offers larger capacity. Examples include external hard drives, portable flash drives, CDs, and DVDs. These devices and media must be either plugged in or inserted into a computer in order to be accessed by the system. Since secondary storage technology is not always connected to the computer, it is commonly used for backing up data. The term storage is often used to describe secondary memory. Secondary memory stores a large amount of data at lesser cost per byte than primary memory; this makes secondary storage about two orders of magnitude less expensive than primary storage.

There are two main types of semiconductor memory: volatile and non-volatile. Examples of non-volatile memory are 'Flash' memory (sometimes used as secondary, sometimes primary computer memory) and ROM/PROM/E-PROM/EEPROM memory (used for firmware such as boot programs). Examples of volatile memory are primary memory (typically dynamic RAM, DRAM), and fast CPU cache memory (typically static RAM, SRAM, which is fast but energy-consuming and offer lower memory capacity per area unit than DRAM). Non-volatile memory technologies in Si-based electronics date back to the 1990s. Flash memory is widely used in consumer electronic products such as cell phones and music players and NAND Flash-based solid-state disks (SSDs) are increasingly displacing hard disk drives as the primary storage device in laptops, desktops, and even data centers. The integration limit of Flash memories is approaching, and many new types of memory to replace

conventional Flash memories have been proposed.

The rapid increase of leakage currents in Silicon CMOS transistors with scaling poses a big challenge for the integration of SRAM memories. There is also the case of susceptibility to read/write failure with low power schemes. As a result of this, over the past decade, there has been an extensive pooling of time, resources and effort towards developing emerging memory technologies like Resistive RAM (ReRAM/RRAM), STT-MRAM, Domain Wall Memory and Phase Change Memory (PRAM). Emerging non-volatile memory technologies promise new memories to store more data at less cost than the expensive-to-build silicon chips used by popular consumer gadgets including digital cameras, cell phones and portable music players. These new memory technologies combine the speed of static random-access memory (SRAM), the density of dynamic random-access memory (DRAM), and the non-volatility of Flash memory and so become very attractive as another possibility for future memory hierarchies. The research and information on these Non-Volatile Memory (NVM) technologies has matured over the last decade. These NVMs are now being explored thoroughly nowadays as viable replacements for conventional SRAM based memories even for the higher levels of the memory hierarchy. Many other new classes of emerging memory technologies such as transparent and plastic, three-dimensional (3-D), and quantum dot memory technologies have also gained tremendous popularity in recent years.

However, it must be noted that the reliability/endurance issues and the read/write latencies required for the higher level memories has defied the use of such NVM options for performance-critical applications. Thus, a direct drop-in replacement of SRAM by NVMs is not possible. Besides, NVM designs are also improving rapidly and it is crucial to take into account the memory cell/array characteristics when designing and proposing architectural modifications. The aim of this PhD is to explore the advantages of using hybrid on-chip memory systems, composed of SRAM and NVM modules, in the context of high performance and low power embedded systems. Both, system architecture design and application to memory mapping are to be addressed in the proposed PhD work, focusing the search in the multimedia, wireless and the Internet of Things (Iot) application domains. We look at two different platforms: Ultra low power and General Purpose. Both the instruction memory and data memory hierarchy are explored for potential NVM inclusion. We propose novel cross-layer design methods for feasible NVM based memories. Our designs address the adverse effects due to the NVM latency and reliability/endurance issues by means of micro-architectural modifications and aims to remove the performance penalty. We also tinker with profiling the applications and optimizing their code to achieve our goal. The area and energy costs are explored when considering our target, design methodology and the specific code optimizations.

According to our simulations, the appropriate tuning of selective architectural parameters of the memory in our proposal and suitable optimizations can reduce the performance penalty introduced by the NVM significantly. Thus, it is worthwhile to pursue explorations on the replacement of SRAM by NVMs at the highest levels of the memory hierarchy. The designing is kept more or less generic in order to facilitate re-use and exploration of other NVMs in place of ReRAM in the future. Overall, we comprehensively evaluate the cost, ability and feasibility of exploring NVM based high level memory systems that can effectively compete with more traditional/SRAM based counterparts. Finally, we present a case study wherein we develop a complete framework (from the technology all the way up to the system level) to analyse the effectiveness of NVM based higher level memories. The study clearly indicates that only through close co-optimization across all the different layers of abstraction can a realistic picture be obtained on emerging NVM technologies. This PhD thesis aims to investigate the feasibility of this rapidly developing new class of memory technologies and scaling of scientific procedures from a system and micro-architecture level point of view.

Resumen

En el campo de la informática, el término 'memoria' se refiere generalmente a dispositivos que son usados para almacenar información que posteriormente será usada en diversos dispositivos, desde computadoras personales (PC), móviles, dispositivos inteligentes, etc. La memoria principal del sistema se utiliza para almacenar los datos e intstrucciones de los procesos que se encuentre en ejecución, por lo que se requiere que funcionen a alta velocidad (por ejemplo, DRAM). La memoria principal está implementada habitualmente mediante memorias semiconductoras direccionables, siendo DRAM y SRAM los principales exponentes. Por otro lado, la memoria auxiliar o secundaria proporciona almacenaje (para ficheros, por ejemplo); es más lenta pero ofrece una mayor capacidad. Ejemplos típicos de memoria secundaria son discos duros, memorias flash portables, CDs y DVDs. Debido a que estos dispositivos no necesitan estar conectados a la computadora de forma permanente, son muy utilizados para almacenar copias de seguridad. La memoria secundaria almacena una gran cantidad de datos a un coste menor por bit que la memoria principal, siendo habitualmente dos órdendes de magnitud más barata que la memoria primaria.

Existen dos tipos de memorias de tipo semiconductor: volátiles y no volátiles. Ejemplos de memorias no volátiles son las memorias Flash (algunas veces usadas como memoria secundaria y otras veces como memoria principal) y memorias ROM/PROM/EPROM/EEPROM (usadas para firmware como programas de arranque). Ejemplos de memoria volátil son las memorias DRAM (RAM dinámica), actualmente la opción predominante a la hora de implementar la memoria principal, y las memorias SRAM (RAM estática) más rápida y costosa, utilizada para los diferentes niveles de cache. Las tecnologías de memorias no volátiles basadas en electrónica de silicio se remontan a la década de 1990. Una variante de memoria de almacenaje por carga denominada como memoria Flash es mundialmente usada en productos electrónicos de consumo como teléfonia móvil y reproductores de música mientras NAND Flash solid-state disks (SSDs) están progresivamente desplazando a los dispositivos de disco

duro como principal unidad de almacenamiento en computadoras portátiles, de escritorio e incluso en centros de datos.

En la actualidad, hay varios factores que amenazan la actual predominancia de memorias semiconductoras basadas en cargas (capacitivas). Por un lado se está alcanzando el límite de integración de las memorias Flash, lo que compromete su escalado en el medio plazo. Por otra parte, el fuerte incremento de las corrientes de fuga de los transistores de silcio CMOS actuales, supone un enorme desafío para la integración de memorias SRAM. Asimismo, estas memorias son cada vez más susceptibles a fallos de lectura/escritura en diseños de bajo consumo. Como resultado de estos problemas, que se agravan con cada nueva generación tecnológica, en los últimos años se han intensificado los esfuerzos para desarrollar nuevas tecnologías que reemplacen o al menos complementen a las actuales. Los transistores de efecto campo eléctrico ferroso (FeFET en sus siglas en inglés) se consideran una de las altenartivas más prometedores para sustituir tanto a Flash (por su mayor densidad) como a DRAM (por su mayor velocidad), pero aún está en una fase muy inicial de su desarrollo. Hay otras tecnologías algo más maduras, en el ámbito de las memorias RAM resistivas, entre las que cabe destacar ReRAM (o RRAM), STT-RAM, Domain Wall Memory y Phase Change Memory (PRAM).

Algunas de estas nuevas tecnologías no volátiles prometen un coste de almacenamiento menor que las actuales tecnologías capacitivas, mejorando al mismo tiempo el resto de factores: combinan la velocidad de static random-access memory (SRAM), la densidad de dynamic random-access memory (DRAM), y la no volatibilidad de la memoria Flash por lo que se han convertido en otra posibilidad muy atractiva para las futuras jerarquías de memorias.

La investigación y desarrollo de estas tecnologías de memorias no volátiles (NVM) ha madurado durante esta última década, hasta el punto de que se considera que serán una reemplazo viable para las memorias basadas en SRAM incluso en los niveles más altos de la jerarquía de memorias. Muchos otras nuevas clases de novedosas tecnologías de memorias como las transparentes y plásticas, 3D y de punto cuántico han ganado también popularidad en los últimos años.

Sin embargo, como no podría ser de otra manera, las tecnologías NVM tienen sus propios inconvenientes y ninguna da las alternativas está realmente hoy en día en disposición de dar el salto para ser una alternativa competitiva a DRAM y SRAM: problemas de fiabilidad/resistencia y de latencia de en lectura/escritura impiden que el remplazo, especialmente de memorias SRAM, sea posible. Por otro lado, los diseños de memorias NVM están mejorándose rápida y es por ello crucial tener en cuenta las características célula/matriz cuando se diseña y se proponen modificaciones en su arquitectura.

El objetivo de este trabajo de doctorado es explorar las ventajas del uso de sistemas de memoria hibridos on - chip, compuestos por modulos SRAM y NVM, tanto en un contexto de alta eficiencia (ratio rendimiento-consumo) como en sistemas de baja consumo energético. En la exploración hemos tenido en cuenta tanto la arquitectura del sistema como la adaptación de las aplicaciones a la nueva jerarquía de memoria, centrando el ámbito de aplicación en búsqueda multimedia, conectividad wireless e 'Internet of Things' (IoT).

Para el estudio se ha considerado tanto la jerarquía de memoria de instrucciones como la de datos, comprobando las particularidades de cada uno de ellos al introducir memorias NVM. Tras constatar que el mero remplazo de memorias SRAM por versiones NVM no es suficiente, presentamos diferentes métodos de diseño cross-layer para hacer uso eficiente de las jerarquías híbridas.

Dichos diseños están orientados a mitigar los efectos adversos provocados por los problemas de latencia y de fiabilidad/resistencia mediante modificaciones de la micro-arquitectura de las memorias y de la arquitectura del sistema. Asimismo se estudia el impacto de diferentes optimizaciones de código específicas para este contexto, que permiten alcanzar el objetivo marcado: minimizar el uso de tecnología SRAM sin impacto en el rendmiento y con beneficios en aspectos como el consumo energético o el área utilizada.

De acuerdo a nuestras simulaciones, el ajuste apropiado de parámetros específicos de las arquitecturas de memoria propuestas, hacen que éstas sean viables y puedan paliar las ineficiencias introducidas por el componente NVM incluso en los niveles más cercanos al procesador de la jerarquía de memoria. Lógicamente, este trabajo no hace más que abrir un camino del que aún falta mucho por recorrer, pues hay más factores, además de los aquí estudiados, que amenzan la viabilidad de nuestras propuestas; pero los resultados son prometedores e invitan a continuar invirtiendo esfuerzos en esta dirección.

Beknopte samenvatting

In de elektronica en informatica, de term 'geheugen' in het algemeen verwijst naar componenten of apparatuur die worden gebruikt om informatie op te slaan die we gebruiken in vele toepassingen, varierend van onze PC's over draagbare apparaten tot slimme sensor- en webtoepassingen. Het primaire/hoofdgeheugen wordt gebruikt voor opslagsystemen die functioneren aan een hoge snelheid (dwz RAM). Het primaire geheugen wordt vaak geassocieerd met adresseerbare halfgeleidergeheugens, namelijk geintegreerde schakelingen bestaande uit silicium gebaseerde transistors. Die laatste worden dan bijvoorbeeld gebruikt als het primaire geheugen, maar ook voor andere doeleinden in computers en andere digitale elektronische apparaten. Het secundaire/extra geheugen biedt programma- en dataopslag die langzamer is om toegang te krijgen, maar het biedt een potentieel veel grotere capaciteit. Voorbeelden hiervan zijn externe harde schijven, draagbare flash drives, CD's en DVD's. Deze apparaten en media moeten ofwel aangesloten worden of in een computersysteem opgenomen worden. Omdat secundaire opslagtechnologie niet altijd verbonden is met de computer, wordt dit algemeen ook gebruikt voor back-up data. Secundaire geheugen slaat een grote hoeveelheid data op tegen lagere kosten per byte dan deze van het primaire geheugen. Dit maakt secundaire opslag ongeveer twee grootteordes minder duur dan primaire opslag.

We kunnen twee hoofdtypen halfgeleidergeheugens onderscheiden: vluchtig en niet-vluchtig. Voorbeelden van niet-vluchtig geheugen zijn flashgeheugen (soms gebruikt als secundaire maar soms ook als primair computergeheugen) en ROM/PROM/EPROM/EEPROM geheugen (voor firmware zoals het opstartprogramma). Voorbeelden van vluchtige geheugen zijn primaire geheugen (meestal dynamische RAM of DRAM), en de snelle CPU cachegeheugens (meestal statische RAM of SRAM). Deze laatste zijn snel, maar kosten meer energie en ze bieden lagere geheugencapaciteit per oppervlakteeenheid dan DRAM. Niet-vluchtig geheugen technologieen in Si-gebaseerde elektronica dateren uit de jaren 1990. Ferro-elektrisch veldeffecttransistor (FeFET) was een van de meest veelbelovende opties ter vervanging van de conventionele

flashgeheugens, die fysieke schalingsbeperkingen hebben op dit momenten. Een variant van ladingsopslaggeheugen aangeduid als Flashgeheugen wordt op grote schaal gebruikt in elektronische producten, zoals mobiele telefoons en muziekspelers terwijl NAND Flash-gebaseerde "solid-state disks"(SSD's) in toenemende mate gebruikt worden om harde schijven te vervangen als de primaire opslagcomponent in laptops, desktops, en zelfs data-centers. De integratielimiet van Flash geheugens nadert, en veel nieuwe soorten geheugens zijn voorgesteld die potentieel de conventionele Flash geheugens vervangen.

De snelle toename van lekstromen in geschaalde silicon CMOS transistoren vormt een grote uitdaging voor de integratie van SRAM geheugens. Er is ook het geval van de gevoeligheid voor lees/schrijfmislukking met laagvermogen regelingen. Als gevolg van deze problemen is de afgelopen tien jaar een sterke bundeling van tijd, middelen en inspanningen gestart in de richting van de ontwikkeling van nieuwe geheugentechnologieen zoals Resistive RAM (ReRAM/RRAM), STT-MRAM, Domain Wall Memory en Phase Change Memory (PRAM). Opkomende niet-vluchtig geheugen technologieen beloven nieuwe richtingen om meer gegevens op te slaan tegen lagere kosten dan de dure siliciumchips die nu gebruikt worden door de populaire consument gadgets zoals digitale camera's, mobiele telefoons en draagbare muziekspelers. Deze nieuwe geheugentechnologieen combineren in theorie de snelheid van statisch willekeurig toegankelijk geheugen (SRAM), de dichtheid van dynamic random access memory (DRAM), en de niet-vluchtigheid van flashgeheugen. Deze zijn sterk onderzocht omdat deze combinatie van eigenschappen zeer aantrekkelijk is voor toekomstige geheugenhierarchieen. Het onderzoek en de informatie over deze niet-vluchtige geheugen (NVM) technologie heeft gerijpt in het afgelopen decennium. Deze NVM worden nu tegenwoordig grondig onderzocht als levensvatbare vervanging voor conventionele SRAM gebaseerde geheugens maar ook voor de hogere niveaus van de geheugenhierarchie. Veel andere nieuwe klassen van opkomende geheugen technologieen zoals transparante en kunststof, drie-dimensionale stapeling (3-D), en de quantum dot geheugentechnologieen zijn ook immens populair in de afgelopen jaren. Wel moet worden opgemerkt dat de betrouwbaarheid/duurzaamheidsproblemen en de lees/schrijfkop latenties een hoger niveau geheugens vereisen voor het gebruik van dergelijke NVM opties in de uitvoering applicaties. Dus een directe vervanging van SRAM door NVM is niet mogelijk.

Daarnaast worden NVM ontwerpen ook snel verbeterd en is het cruciaal om rekening te houden met de geheugencel/-matrix bij het ontwerpen en het voorstellen van architecturale modificaties. Het doel van dit doctoraat is om de voordelen van het gebruik van hybride on-chip geheugensystemen, bestaande uit SRAM en NVM modules te verkennen, in de context van systemen met zowel hoge prestaties als ingebedded systemen met (ultra)laag

stroomverbruik. Zowel het systeemarchitectuur ontwerp als de toepassing in het geheugen worden in kaart gebracht in het voorgestelde doctoraat, met de nadruk op het zoeken in de domeinen multimedia, draadlozenetwerken en het internet van dingen (Iot). We kijken naar twee verschillende platformen: Ultra low-power en General-Purpose. Zowel de instructiegeheugen en data geheugenhierarchie worden onderzocht op mogelijke NVM integratie. Wij stellen nieuwe crossdisciplinaire ontwerpmethoden voor, vooral voor haalbaar NVM gebaseerd geheugenopslag. Onze ontwerpen richten zich op de negatieve effecten als gevolg van de NVM latentie en de betrouwbaarheid/duurzaamheid vraagstukken door middel van micro-bouwkundige aanpassingen. Het heeft als doel om de vermindering van de prestaties te vermijden of the verwijderen. We hebben ook gesleuteld aan het profileren van de applicaties en het optimaliseren van hun code om ons doel te bereiken. De kosten en energie in elk gebied worden onderzocht bij het overwegen van ons doel, de ontwerpmethodologie en de specifieke codeoptimalisaties. Op basis van onze simulaties kan de juiste afstemming gebeuren van selectieve architecturale parameters in de geheugenorganisatie. Ons voorstel laat dus toe door middel van geschikte optimalisaties de prestaties van NVM aanzienlijk te verminderen. Het is dus de moeite waard om verkenningen over de vervanging van SRAM na te streven door NVM's op het hoogste niveau van het geheugen hiararchie. Het ontwerpen wordt min of meer generiek gehouden met het oog op hergebruik en de verkenning van andere NVM's in plaats van ReRAM in de toekomst te vergemakkelijken. Kortom, we evalueren uitvoerig de kosten, het vermogen en de haalbaarheid van het verkennen van NVM-based hoog-niveau geheugensystemen die effectief kunnen concurreren met meer traditionele/SRAM gebaseerde tegenhangers. Dit proefschrift heeft als doel de haalbaarheid te bestuderen van deze zich snel ontwikkelende nieuwe klasse van geheugentechnologieen en de schaalvergroting van de wetenschappelijke procedures van een systeem en micro-architectuur niveau oogpunt te onderzoeken.

Chapter 1

Introduction

1.1 Context and Motivation

1.1.1 Global Context

It is becoming abundantly clear that portable electronic devices and mobile systems have now become a key part of everyday life. This is also easily validated by market research on portable media players, gaming consoles, smartphones and more recently wearables (Figure 1.1). Essentially, we want entire libraries and databases of information in the palm of our hand. Novel device concepts with new physical operating principles are now required to match the consumer demands. Consumers expect ever more performance and features from these devices, while the battery capacity hardly improves. The growing trend of ubiquitous computing and being 'always-online' are also imposing new challenges on the design of these embedded mobile systems. These systems and devices are expected to optimally support multiple applications and their heterogeneous execution profiles, while meeting aggressive performance, energy, thermal and cost requirements.

A low energy consumption per function is quite naturally one of the most fundamental requirements in their design. These devices typically operate with a clock frequency between 100MHz and 1GHz. The importance of a low energy consumption per operation is, however, not limited to these classical portable devices. Even in high-performance systems such as powerful desktop processors, digital signal processors (DSPs) and database servers energy efficiency is a key issue. A lower energy consumption allows the manufacturer to use less

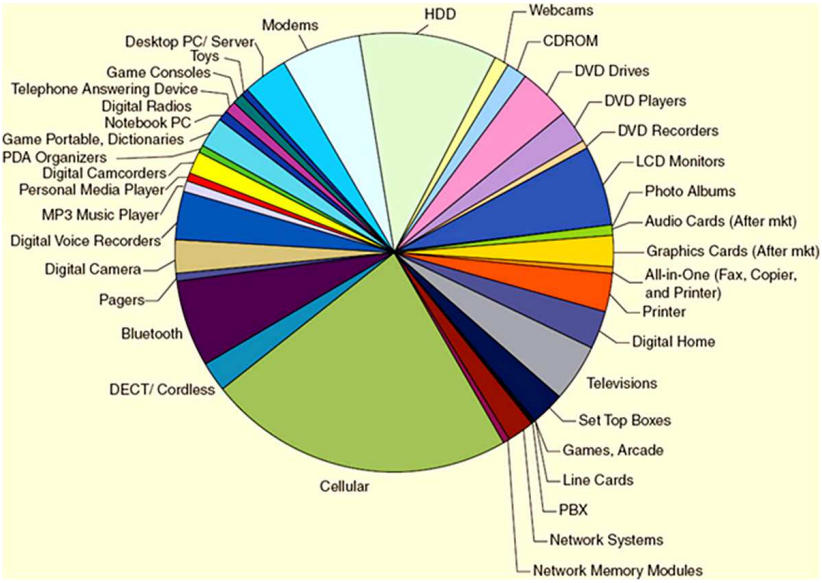


Figure 1.1: The different NVM applications in the electronics industry with respect to market size in 2012. Reprinted from [131] with permission.

expensive packages and cooling solutions[31] and reduces the energy bill for the customer. Today, the clock frequency for these devices is typically in the range from 500MHz to 4GHz. Designing systems under such tight performance, energy and cost budgets leads to the following major challenges (or 'walls') - **Design Space Complexity Wall**, **Design Efficiency Wall**, **Design Cost Wall**, **Technology Scaling Wall** (Figure 1.2 and 1.3), **Battery Wall** and **Memory Wall**. The bottlenecks defined by these walls have created a breakpoint that presents a great opportunity for innovation and alternative technologies.

In most of these domains and systems, memories consume a significant part of the total energy budget [207, 111]. This is certainly true for the very data intensive contemporary multimedia applications [130], but similar conclusions apply to low power sensor nodes [144] and in-fact for all familiar computing platforms ranging from hand-held devices to large supercomputers. After optimizations have been applied, most applications have strong data locality. This increases the relative importance of the small memories that are closest to the processor to such an extent that they often dominate the energy consumption. Hence, energy efficient implementations for these relatively small memories are a key requirement to enable further extensions of the capabilities

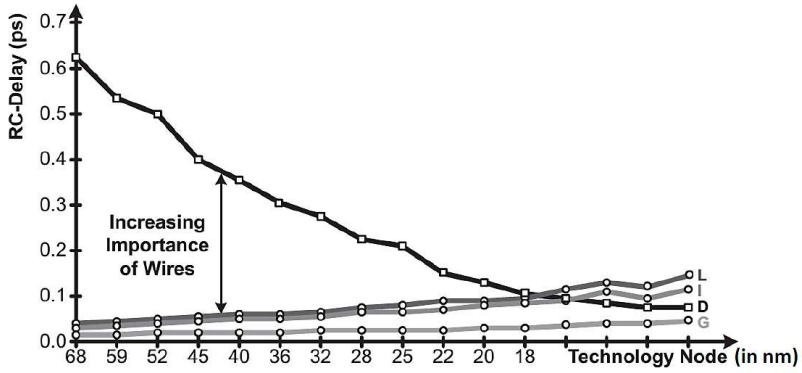


Figure 1.2: ITRS prediction of increasing RC-delay of local (L), intermediate (I) and global (G) interconnects as compared to logic devices (D) with the scaling of technology nodes (in nm) [48]

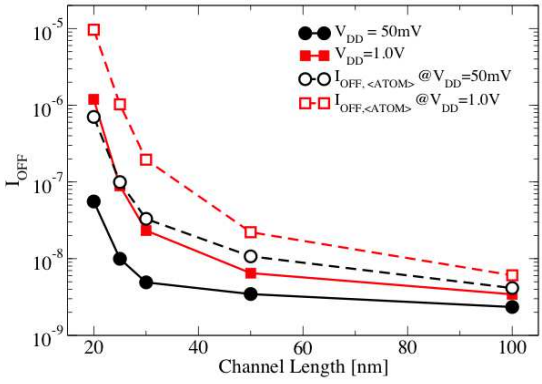


Figure 1.3: Transistor scaling [83]

of these devices. Until recently, the energy consumption per operation of both memories and processors improved on a regular basis thanks to the ongoing miniaturization of technology. Smaller minimal dimensions not only allowed to put more transistors onto the same area, but also enabled higher speeds and lower energy consumption per operation. In this era of 'happy scaling' [40], the energy per operation reduced in proportion to λ^3, with λ the smallest feature size that can be realized in the technology. If a design is re-scaled from a 0.5μm technology (1990) to a 130nm technology (2003), the energy consumption per operation is reduced with a factor of 50, without significant effort from the designer. All this ended with the introduction of the 90nm technology node and we hit the '**Technology Scaling Wall**' [40, 53] (Fig. 1.3).

Because leakage currents started to dominate the energy budget, the threshold voltage could no longer be reduced (Fig.1.3). This in turn made it impossible to further reduce the supply voltage without taking a significant performance hit. The variations between adjacent transistors, also known as mismatch, increased quickly with smaller dimensions. This mismatch has a detrimental impact on the available noise margins for memory cells [14], which further impedes the reduction of the power supply. Another important problem for technology scaling is that the performance of wires does not scale as well as that of transistors [11]. Hence, wires have become a first-order design consideration, especially in memory design. High-performance systems might still benefit from further miniaturization, at least down to the 7nm node. However, the energy per operation now reduces proportional to λ at best, rather than λ^3. For systems with a lower activity, the potential advantage is less clear due to the larger impact of leakage currents and transistor variations.

1.1.2 The Memory Perspective

For some applications, further miniaturisation can account for a part of this reduction, but only if new circuit and system solutions are devised to cope with the increasing variability and leakage currents. Even though scaling seems to become less effective and less affordable for many applications, technology can still have a tremendous impact on system performance by offering new integrated technology options such as efficient inductors, high density capacitors, new transistor architectures and new cell types for volatile and non-volatile memory and data storage. Currently, most of the memory systems and storage are dominated by SRAM(Fig. 1.4), DRAM(Fig. 1.5) or Flash(Fig. 1.6). SRAM memories face a number of issues like increasing sub-threshold leakage with scaling and susceptibility to read/write failure with dynamic voltage scaling schemes or a low supply voltage (V_{dd}) [64, 149]. The low V_{dd} in turn leads leads to insufficient yield. In DRAM (3-D

DRAM), the aggressive scaling of the cell capacitance threatens the sensing margin and obtaining enough data retention characteristics [157]. As the cell transistor having the buried gate (BG) structure is scaled down, the off-state characteristics is deteriorated [135]. This phenomenon is attributed to both the weakened gate control-ability of the body-tied fin transistor and the increased interference between cell transistors. The challenge for 3D MLC NAND Flash arising due to aggressive scaling and the resulting limited number of stored electrons is retention failure . Due to a block-based architecture, the random access time for NAND Flash to any given bit tends to be slow (25μs), with a significantly faster readout of data blocks ($23 - 37$ MB/s) after this initial access delay [16]. Erasing a block also tends to be extremely slow (2ms).

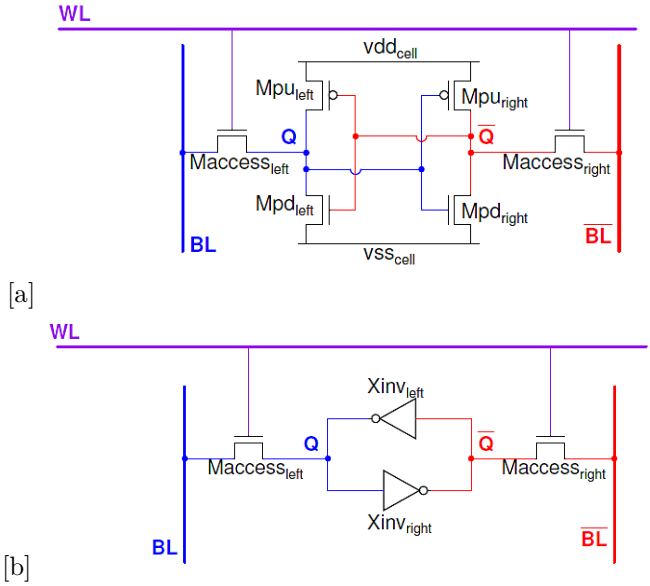


Figure 1.4: Traditional 6T SRAM cell. (a) Transistor schematic. Maccess= access transistor, Mpd=pull-down transistor, Mpu=pull-up transistor. (b) Same cell, different view.

Along with the above mentioned bottlenecks that arise due to the '**Technology Scaling Wall**' for traditional memories, the ever increasing gap between processing and memory will aggravate the situation further in future complex MPSoC (Multi-Processor System on Chip) designs. Mitigation of this memory gap ('**Memory Wall**') will require comprehensive solutions that encompass complex memory hierarchies, integration of emerging memory technologies, innovative system architectures, new programming models, etc. This has

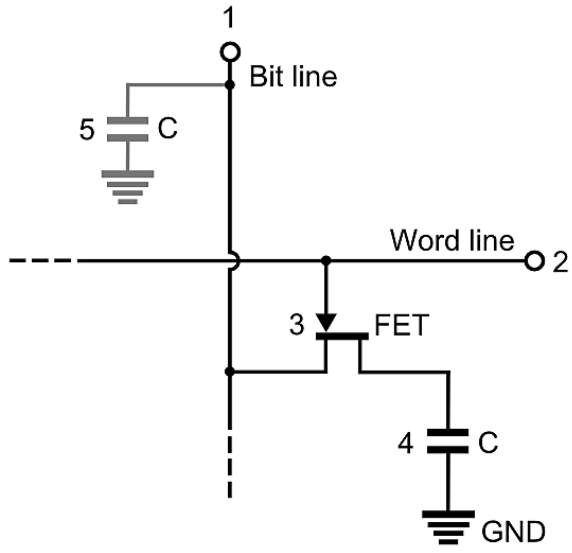
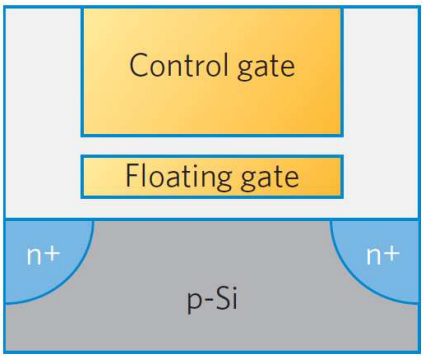


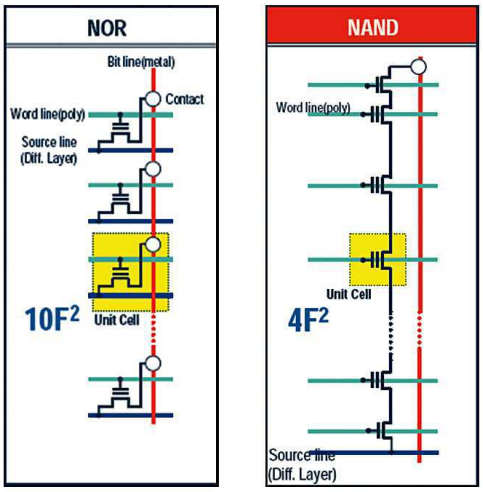
Figure 1.5: DRAM bit-cell

driven the search for the so-called '***Universal Memory***' that can function as both memory and storage [110]. However, the design specifications for memory (volatile data storage, fast, expensive) and for storage (non-volatile data storage, slow, inexpensive) are different, and they often have different data access standards and protocols (Figure 1.7). The possibility of blurring the design boundary between memory and storage, and coming up with new data access modes and protocols that are neither 'memory' nor 'storage' has been looked into for quite some time now. Table 1.1 details the requirements and expected memory characteristics for the different levels of the memory hierarchy.

A memory system is essentially a hierarchy of storage devices with different capacities, costs, and access times/latencies. A proper definition of the hierarchy is important for computer architecture design, algorithm predictions, and also the lower level programming constructs. At the topmost level are the CPU registers, that are used to hold the most frequently used data. These are memory cells built right into the CPU that contain specific data needed by the CPU, particularly the arithmetic and logic unit (ALU). An integral part of the CPU itself, they are controlled directly by the compiler that sends information for the CPU to process. Registers are managed at compile time when the high



[a]



[b]

Figure 1.6: (a) Traditional Flash cell with floating and control gates. (b) Transistor Schematic for NOR and NAND. [131]

level program is converted to machine specific assembly. Once, an 'Instruction Set Architecture' or ISA is defined for a particular family of processors, it is difficult to change parameters such as number and width of registers without breaking compatibility with previously written software. Small, fast cache memories memories or scratchpad memories make up the next level of the memory hierarchy. Cache memories hold a subset of the data and instructions stored in the relatively slow main memory. There can be several sub-levels of cache memory (termed L1, L2, L3). The main memory holds copies of the data stored on large, slow disks (Secondary Storage). Main Memory(RAM) however, only retains it's contents when the power is on, so needs to be stored on more

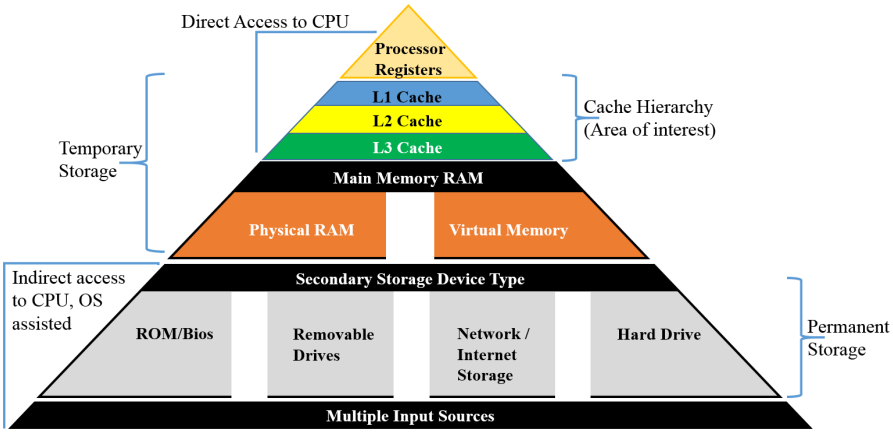


Figure 1.7: Memory Hierarchy and corresponding design specifications across the different layers

Table 1.1: Requirements of the different levels of the Memory hierarchy

Parameters	**Caches / Scratchpad**	**Main Memory**	**Storage**
Performance	Very fast (0.5-8ns)	fast (5-10ns)	slow (∼10ms)
Area	Very small	Large	Very large
Cost/capacity	High	Medium	Cheap
Technology	SRAM	DRAM	Flash

permanent storage. The secondary storage in turn often serve as staging areas for data stored on the disks or tapes of other machines connected by networks. Memory hierarchies work because well-written programs tend to access the storage at any particular level more frequently than they access the storage at the next lower level. So the storage at the next level can be slower, and thus larger and cheaper per bit. The overall effect is a large pool of memory that costs as much as the cheap storage near the bottom of the hierarchy, but that serves data to programs at the rate of the fast storage near the top of the hierarchy.

Indeed, the adoption of Flash in the memory hierarchy (albeit on a separate chip from the processor) inspired the exploration of computing architectures that capitalize on the salient features of Flash: non-volatility and high density. At the same time, new types of non-volatile memory have emerged that can easily

be integrated on-chip with the microprocessor cores because they use a different set of materials and require different device fabrication technologies from Flash [215]. Some of them can be programmed and read quickly; others can have very high data storage density. These memories are free from some the limitations of Flash like low endurance, need for high voltage supply, slow write speed and cumbersome erase procedure. Coincidentally, these new memories store information using new types of physics that do not rely on storing charge on a capacitor as is the case for SRAM, DRAM and Flash. These new ***Universal Memory*** candidates may also allow the monolithic integration of memories and computation units in three-dimensional chips for future computing systems [193].

As a result, significant effort and resources are being spent on developing the emerging memory technologies like in recent years. Owing to their desirable characteristics like low leakage power (with an appropirate selector) and high density, these NVMs are being explored as efficient alternatives even for higher level SRAM memory systems like scratchpad memory and cache [195, 70, 129, 186, 190]. In-fact most of today's computing systems already use a hierarchy of volatile and non-volatile data storage devices to achieve an optimal trade-off between cost and performance [153]. Some of the major nonvolatile memory technologies are: Flash Memory, Ferro-electric random-access memory (FeRAM), Magnetic random-access memory (MRAM), Phase-change memory (PCM/PRAM), and Resistive RAM (RRAM/ReRAM) and more recently Domain Wall Memory (DWM) [16, 131, 215]. These memories are more mature and well defined with respect to heir material properties, physical limitations and feasibility of incorporation into the memory hierarchy. Some of the newer NVM memories like DNA memory, plastic memory and Domain Wall memory are less understood and thus comparatively less mature when it comes to technology development, integration schemes and design technology co-optimization.

System Level Viewpoint

While many of the proposed emerging memories have the potential to break into the market dominated by SRAM, DRAM and Flash, they haven't been explored as extensively and are simply not mature enough. The semiconductor industry will only gradually shift away from the traditional choices once they have hit the absolute limits/walls. This is based on the a number of factors like performance shortcomings, cost of integration and CMOS compatibility. In-order to incorporate these new memories and thus enable a host of new exciting applications, a further radical reduction of energy consumption per operation along while meeting the performance requirements is crucial. Three tracks

can contribute to this energy reduction: improvements in process technology, in circuit design and system-level design. In -short, the technology scaling wall related issues can be effectively mitigated only when the implications of different design choices made at the system-level architecture are understood at the technology design phase and vice-versa. A number of technology innovations and options, both mature and developing have been laid out in the earlier sections. To correctly assess this multitude of possibilities and to use them efficiently requires a deep understanding of their circuit and system-level implications. At the system-level, co-optimization of the algorithms and the hardware system can often result in a huge reduction in the energy consumption. The maximal benefits from this approach are obtained when application and domain specific memories are employed. These memories can make the optimal trade-off between the specifications (e.g. energy, delay, area, yield and special features) for the application domain.

Initial experiments and analysis reveal that direct drop-in replacement of the new NVMs in memory or storage is not feasible without a re-organization of peripheral circuitry, micro-architectural changes and system optimization. Some of the explorations are reported in [112, 113] and they show that the performance, energy, and endurance of PCM chips can be greatly improved with the a combination of circuit, architecture and system modifications and techniques. Similar conclusions have been reached upon evaluation of the complete replacement of DRAM with STT-MRAM [107] reorganization of peripheral circuitry of STT-MRAM chips (with the goal of minimizing the number of writes to the STT-MRAM cell array, as write operations are high latency and high-energy in STT-MRAM) enables an STT-MRAM based main memory to be more energy-efficient than a DRAM-based main memory. In almost all of these NVMs there is a read write asymmetry when it comes to both power and performance. They have low energy, fast reads with high energy, low latency writes. One can achieve more efficient designs of ReRAM/PCM/STT-MRAM chips by taking advantage of the this low power non-destructive nature of reads. This enables improved NVM performance and also increased endurance. The system level requirements and recent progress in this area is detailed in Section 2.4.

1.2 PhD Focus and Objectives

Despite semiconductor progress, till date no single semiconductor nonvolatile memory device at the mass-production stage satisfies all the above constraints. Also, the understanding and development of these emerging NVM technologies is limited. Thus, most hand-held systems are equipped with heterogeneous

non-volatile memory devices combining Flash memories and battery-backed dynamic memories in self-refresh mode. Most of the proposals to overcome the bottlenecks/walls hinted at earlier consist of the emerging NVM technologies being utilized along with the traditional ones in order to limit the negative impacts and maximize the positive ones. As various types of non-volatile memory devices show different characteristics, we must evaluate and analyse them carefully. Traditional architectures and allocation schemes will not be able to guarantee performance requirements and energy efficiency because they are not designed to optimally utilize the characteristics of the emerging memories. The following points list the general shortcomings that are present for the current state of the art NVMs:

- Latency: While read latency may not be an issue for NVMs with a very high on-off ratio, it is an issue for STT-MRAM when we look at L1 and L0 caches. Write latency is an issue for all NVMs
- Lifetime: STT-MRAM with a nominal lifetime of $\sim 10^{12} - 10^{13}$ (and tail upto $\sim 10^{15} - 10^{16}$) has the highest lifetime when it comes to NVMs and even this might not be enough for certain application domains. Other options are severely lacking ([5], [106]).
- Switching current and Selector: At advanced scaling nodes, it is difficult to find selectors that are able to drive enough current/voltage that can switch these NVMs.
- Scalability: Voltage and area scaling of the NVM elements is also hampered by the actual mechanisms that drive the switching phenomenon.

The main goal of this thesis is to study hybrid memory systems that lie closest to the core (referred to in section 2.4.2) at the system architecture abstraction. We exploit components from conventional memory devices that have been proposed in the past years, together with novel emerging non-volatile memory technologies. Within hybrid memory systems, the exploration will be narrowed down to focus on the highest levels of the memory hierarchy: software and hardware controlled cache memories in L1 levels (Scratchpad and Cache L1). From application point of view, both low power and high performance systems will be looked at. The main objectives of the thesis will be to show that system level techniques could (in combination with circuit and cell modifications) effectively mitigate the hindrances for the introduction of the best NVM candidates into the highest levels of the memory hierarchy. The study will also ensure the set-up of a feedback loop based on which process technology and circuit design can be steered to make NVM technology feasible.

Both, system architecture design and application to memory mapping are to be addressed in the PhD work. The instruction and data memory/cache hierarchy are explored for potential ReRAM (OxRAM) and STT-MRAM inclusion. The designing is kept more or less generic in order to facilitate re-use and exploration of different NVMs in the future. A set of typical system-level requirements for the target applications/domain are the following:

- Energy per read/write access should be reduced as much as possible because of the portable context, preferably in the same range as the foreground memory access so around 100fJ/read word (writing happens less frequently so it can be more expensive).
- Area should remain low but due to the limited sizes (16 - 128Kb for L1 and 128Kb - 8Mb for L2) the cell area can be a bit relaxed compared to standalone memories. This is subject to change however, with increasing L1 sizes and changing application spheres.
- Endurance should be high: preferably up to 10^{16} cycles for write, but retention in the L1-L2 layers can be relaxed to a few days or even few hours to trade-off for more endurance. The limited amount of data that has to be stored longer can easily be backed up in an off-chip NVM with long retention with little impact on energy consumption.
- After architectural mitigation and optimizations the resulting read access speed should be below $\sim$1 ns (for general purpose/server contexts), whereas write access speed can tolerate a larger latency but not more than 2 cycles in L1 and 8 cycles in L2 normally.

The asymmetric behaviour of NVM cells and their effects at the system level are studied. Asymmetric interfaces to and from the Processor core are explored in order to deal with the SRAM/NVM disparities in read and write accesses. Memory mapping schemes have to be developed taking into account scheduling, codes that will be used based on the target applications (e.g. loop structures etc), data layout etc for application specific processors. Suitable address frameworks and address compilers have to be found/developed for this purpose. Code optimizations by transformations are also looked into in-order to simplify processes and for energy and access optimizations. Lastly real time applications for the proposed architectures are analyzed. The last stage of the PhD focuses on the cross layer exploration of a NVM technology (based on real silicon measurements) taking into account complete parasitic, with design rule checks so that the design can be implemented on silicon with confidence.

1.3 List of Simulators Used

Over the course of the PhD thesis, a number of simulators have been used at the different levels of abstraction. CACTI and NVSim (referred to in detail in sections 3.6 and 5.3) are memory array simulators that utilize an empirical modeling methodology and only provide a rough abstraction of the memory array. They do not have the capability to run simulations with accurate device models, full parasitic extraction, account for transmission losses and/or check the integrity of circuit designs and to predict circuit behavior. CACTI takes into account parameters that give an overview of the memory array and the technology node (like Cache Size, Line size, Number of Banks, Number of Ports, Technology Node, Transistor type etc) to generate power, timing and area as the output. NVSim has the added functionality of customizing the global memory array configuration and the bit-cell parameters. CACTI is used for the work detailed in chapter 3 and NVSim for the work detailed in chapter 5 and 6.

The TARGET simulator (section 3.4), ADRES framework (section 4.3) and GEM5 (section 5.5) represent system architecture abstractions. These are required to mimic our target exploration space from a memory system architecture viewpoint for different domains. TARGET addresses domain-specific instruction-set processors for the embedded application domain; ADRES addresses coarse-grain array architectures for the wireless domain; and GEM5 addresses general-purpose instruction-set processors for the high-performance domain. Since our aim is to explore and analyze the system impact when introducing new NVM memories, we need to simulate the behavior of representative sets of applications in realistic environments. Hence, we require system level cycle-accurate simulators. They do not simulate signal switching, but assign a given latency to every event. Then they do a per-cycle simulation checking which events happen at that time (issue arithmetic operations, request a load operation to memory system, write back results to the register-file etc) and update the statistics that will be finally present in the output file (including total simulation time, cache and memory behavior...).

1.4 List of Contributions

- Manu Komalan, M. Hartmann, J. I. Gómez, C. Tenllado, A. Artes, and F. Catthoor, '*System level exploration of Resistive-RAM (ReRAM) based hybrid instruction memory organization*', in Memory architecture and organization workshop (MeAOW), 2012.
  Summary follows: ' A SRAM based traditional instruction memory

organization suitable for the target applications is taken as the base and different ReRAM based hybrid organizations are designed as alternatives keeping in mind the ReRAM limitations, and energy and performance trade-offs. Both, dynamic instruction mapping and architectural design changes are utilized to minimize the ReRAM limitations and maximize its positive contributions. Energy and performance values were obtained by extension of CACTI models and VHDL simulations. The ReRAM based hybrid instruction memory organization showed significantly lower energy consumption (up-to 82.07% read energy reduction) even in case of 0% performance penalty. The focus here is on low power embedded systems for wireless or multimedia target applications.'

- F. Catthoor, P. Komalan Manu, and S. Cosemans, '*Method and device to reduce leakage and dynamic energy consumption in high-speed memories*', European Patent 2013014111, US20140289457, Publication Number WO2013014111 A1, Jan. 31, 2013.
  Summary follows: 'The patent presents an invention that relates to a microcomputer comprising a microprocessor unit and a first memory unit. The microprocessor unit comprises at least one functional unit and at least one register. This is a wide register comprising a plurality of second memory units which are capable to each contain one word, said wide register being adapted so that the second memory units are simultaneously accessible by the first memory unit, and at least part of the second memory units are separately accessible by the at least one functional unit. The first memory unit here is an embedded non-volatile memory unit.'

- M. Komalan, J. Gómez, C. Tenllado, M. Hartmann, P. Raghavan, and F. Catthoor, '*Feasibility exploration of NVM based I-cache through MSHR enhancements*', in Design, Automation Test in Europe Conference Exhibition (DATE), 2014. DOI: 10.7873/DATE.2014.034.
  Summary follows: 'The main focus of this paper is the exploration of write delay and write energy issues in a NVM based L1 Instruction cache (I-cache) for an ARM like single core system. We propose a NVM I-cache and extend its MSHR (Miss Status Handling Register) functionality to address the NVMs write related issues. According to our simulations, appropriate tuning of selective architecture parameters can reduce the performance penalty introduced by the NVM ($\sim$45%) to extremely tolerable levels ($\sim$1%) and show energy gains up to 35%. Furthermore, on configuring our modified NVM based system to occupy area comparable to the original SRAM-based configuration, it outperforms the SRAM baseline and leads to even more energy savings.'

- M. Komalan, J. Gómez, C. Tenllado and F. Catthoor, '*Exploration of NVMs for Low Level Memories*', in Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems (ACACES), Poster Session, 2014.
  Summary follows: 'The main focus here is the exploration of NVMs for higher levels of the memory hierarchy, keeping in mind their limitations. Two different platforms are analyzed to check the feasibility of SRAM replacement at higher levels. One of our targets is the instruction memory in low power embedded systems and the other is the L1 Instruction cache (I-cache) in a general purpose ARMsystem. According to our simulations, architectural changes and appropriate tuning of selective parameters can reduce the performance penalty introduced by the NVM to extremely tolerable levels and also show energy reduction.'

- Manu P. Komalan, M. Hartmann, J. I. Gómez, C. Tenllado, A. Artes, José Francisco Tirado Fernández and F. Catthoor, '*Design exploration of a NVM based hybrid instruction memory organization for embedded platforms*', in Design Automation of Embedded Systems (DAES), Springer, 2014. DOI: 10.1007/s10617-014-9151-8.
  Summary follows: 'The main focus of this paper is the exploration of a NVM based scratchpad memory in ultra low power embedded systems for wireless or multimedia target applications. A SRAM based traditional instruction memory organization suitable for the target applications is taken as the base. Different Resistive RAM based organizations are then designed as alternatives keeping in mind their limitations (write process related), and energy and performance trade-offs. The NVM array design is explored and optimized based on energy and performance trade-offs. Dynamic instruction mapping and architectural design changes are utilized to minimize ReRAM limitations and maximize its positive contributions. Energy and performance values are obtained by extension of CACTI models, Spice and VHDL simulations. The best ReRAM based hybrid instruction memory organization that utilizes our proposed methodology showed significantly lower energy consumption even in case of 0% performance penalty.'

- M. Komalan, J. Gómez, C. Tenllado, José Francisco Tirado Fernández and F. Catthoor, '*System Level Exploration of a STT-MRAM based level 1 Data-Cache*', in Design, Automation Test in Europe Conference Exhibition (DATE), 2015. ISBN: 978-3-9815370-4-8.
  Summary follows: 'Since Non-Volatile Memory (NVM) technologies are being explored extensively nowadays as viable replacements for SRAM based memories in LLCs and even L2 caches, we try to take stock of their potential as level 1 (L1) data caches. A direct drop-

in replacement of SRAM by NVMs is, however, still not feasible due to a number of shortcomings with latency (write or read) and/or endurance/reliability among them being the major issues. With advancements in cell technology, and taking into account the stringent reliability and performance requirements for advanced technology nodes, the major bottleneck to the use of STT-MRAM in high level caches has become read latency (instead of write latency as previously believed). The main focus of this paper is the exploration of read penalty issues in a NVM based L1 data cache (D-cache) for an ARM like single core general purpose system. We propose a design method for the STT-MRAM based D-cache in such a platform. This design addresses the adverse effects due to the STT-MRAM read penalty issues by means of micro-architectural modifications along with code transformations. According to our simulations, the appropriate tuning of selective architecture parameters in our proposal and suitable optimizations can reduce the performance penalty introduced by the NVM (initially 54%) to extremely tolerable levels (8%).'

- P. Komalan Manu, M. Hartmann, J. I. Gómez, C. Tenllado and F. Catthoor, '*Low-Layer Memory for a Computing Platform*', European Patent EP15186601.9 (Filed), Sept. 24, 2015.
  Summary follows: 'The invention relates to a memory hierarchy having at the very least a Level 1, memory structure comprising a non-volatile memory unit as L1 data memory and a buffer structure (L1-VWB). The buffer structure comprises a plurality of interconnected registers with an asymmetric organization, wider towards tge non-volatile memory unit than towards a data path connectable to processor. The buffer structure and said non-volatile memory unit are arranged for being directly connectable to the processor so that data words are selectable to be read or written by the processor.'

- Manu Komalan, Sushil Sakhare, Trong Huynh Bao , Siddharth Rao, Woojin Kim, Christian Tenllado, José Ignacio Gómez, Gouri Sankar Kar, Arnaud Furnemont and Francky Catthoor, '*Cross-layer design and analysis of a low power, high density STT-MRAM for embedded systems*', in IEEE International Symposium on Circuits and Systems (ISCAS), 2017 (Accepted).
  Summary follows: 'STT-MRAM (Spin Transfer Torque Magnetic Random Access Memory) has attracted considerable attention of late since it is the most promising logic compatible nonvolatile memory that is suitable for advanced logic nodes (N28 and beyond) in terms of endurance, speed and power. Embedded STT-MRAM has thus been proposed as a candidate for emerging low standby power connectivity systems such as

IoT (Internet-of-Things) and wearables. We utilize the high performance CoFeB based perpendicular MTJ (pMTJ) device to realize a low power and highly dense array for such embedded systems. This study is carried out on the TSMC 28nm technology node and includes a complete cross-layer design and analysis framework ranging from device modeling to circuit design, layout and system implementation. The Process Variations and Temperature (PT) impact of the MTJ on the STT-MRAM design and correspondingly the total energy consumption and performance of the system is also analyzed. We report a ∼85% reduction in the energy consumption compared to the baseline SRAM based system for near negligible performance penalty (<5%)'.

1.5 PhD structure

The remaining thesis is structured as follows:

Chapter 2 presents the state of the art contributions with respect to the field of NVMs both from the technology and system level perspective.

Chapter 3 presents a design methodology at the circuit and architecture abstraction levels for a Resistive RAM based instruction memory organization akin to a scratchpad. The exploration is focused on ultra low power embedded instruction memories for low power wireless/multimedia applications with loop dominated codes. Both, dynamic instruction mapping and architectural design changes are utilised to minimize the ReRAM limitations and maximize its positive contributions.

Chapter 4 highlights the extension of the design methodology for ReRAM incorporation into instruction memories for a custom Coarse-Grained Reconfigurable Architecture (CGRA) Framework based on the ADRES platform. Here, an assymetric ReRAM/VWR(Very Wide Register) based interface akin to the one in chapter 2 is presented along with a suitable memory access scheme that effectively deals with these issues.

Chapter 5 details the exploration of write delay and write energy issues in a NVM based L1 Instruction cache (Icache) for an ARM like single core general purpose system. A STT-MRAM based I-cache and extend its MSHR (Miss Status Handling Register) functionality to address the NVMs write related issues.

Chapter 6 presents variations of an STT-MRAM based D-cache. The more write intensive nature of Data caches introduce a challenge and we explore different micro-architectural modifications to overcome them. These

proposals are comparatively analysed. A systematic approach to facilitate code transformations and optimizations is employed on top of these architectural modifications.

Chapter 7 concludes thesis with a short summary of the important contributions laid out in the thesis, their implications and the future outlook/challenges from a system point of view. A short summary of the case study that features the complete implementation of a STT-MRAM based cache memory organization for a target an ultra low power platform, and serves as the concluding touch for our explorations is also presented (Section 7.1.5). This features development of a compact model of the device based on silicon data, circuit design based on an accurate commercial technology node and it's considerations along with design rule checks. The output of the case study reinforces our earlier assumptions from abstractions and helps paint a complete picture on the feasibility of NVMs for the highest levels of the memory hierarchy.

A flowchart highlighting the various chapters along with their dependencies and inter-relationships is highlighted in Figure 1.8

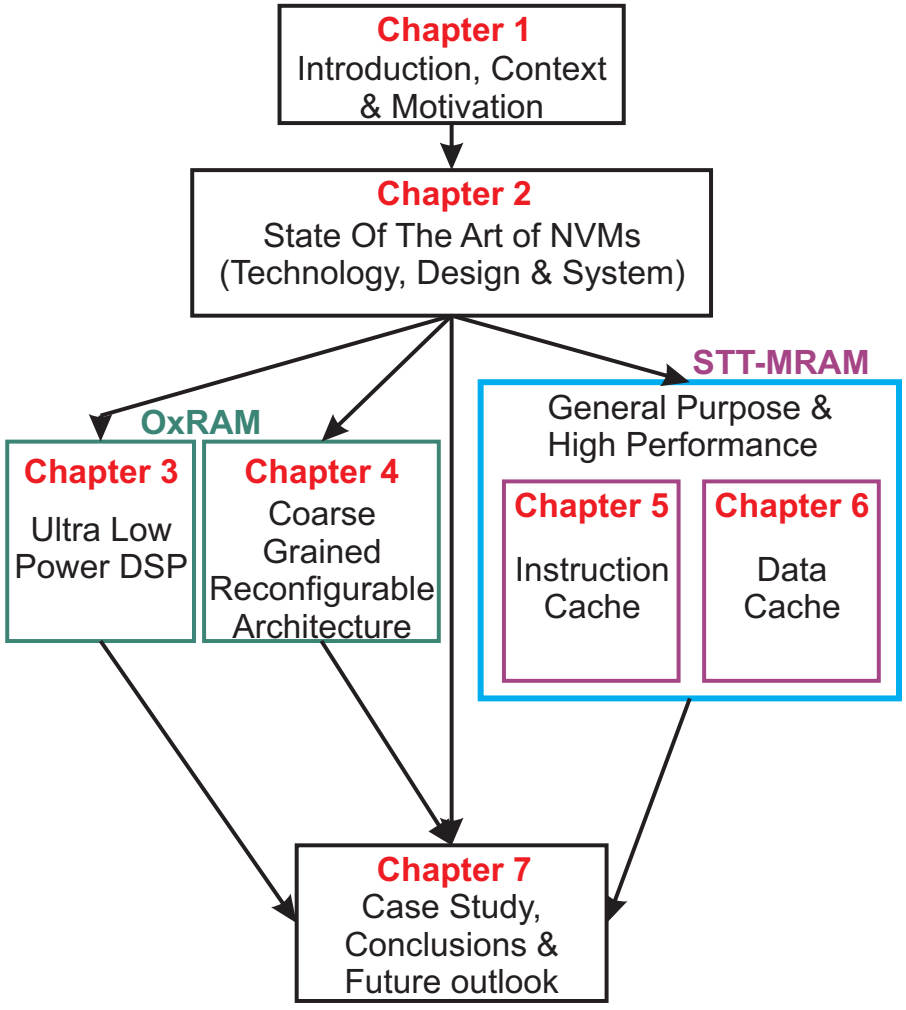


Figure 1.8: Dependency flowchart.

Chapter 2

NVMs: State of the Art

2.1 General Overview of Non Volatile Memories

While this PhD thesis is focused on the highest/higher levels of the memory hierarchy (closest to the processor cores like Scratchpad/L1 memories and caches), it is worthwhile to provide a general overview of the NVM technologies (both mature and emerging) with state of the art comparisons. This is important to understand, identify and classify NVMs based on the different physical mechanisms behind their operation and their positions in the memory hierarchy. A number of these emerging NVMs can also fill the wide gap between Main Memory and Storage. These NVMs are storage class memories and can be roughly classified as SCM-M (SCM - Memory) and SCM-S (SCM - Storage) based on their proximity to Main Memory or Storage. The design space explorations and system level analysis carried out over the course of the thesis can also be motivated better based on the potential characteristics and limitations of these NVMs. Section 2.2 details the device characteristics for the NVMs (both mature and emerging), the state of the art with regards to their development. Section 2.3 briefly touches upon other new memory technologies that are in very early development stages and also provides tables that present the most salient device characteristics of conventional memory technologies along with mature and emerging NVM ones. In section 2.4.2, hybrid memory systems are touched upon by means of the state of the art and classification on the basis of the place in the memory hierarchy.

2.2 Overview of NVM device characteristics

2.2.1 Flash Memory

The possibility of a Non-Volatile Memory device was first conceived with the arrival of the Flash memory. The idea of using a floating gate (FG) device to realize a NVM device was suggested for the first time in 1967 by Kahng D and Sze SM at Bell Labs [84]. 'Flash' memory, is characterized by a large-block (or 'sector') erasing mechanism. Since the advent of portable electronic devices such as music players and mobile phones, however, Flash memory, was introduced into the information storage hierarchy between the DRAM and the HDD. Flash has now become the dominant data storage device for mobile electronics; increasingly, even enterprise-scale computing systems and cloud data storage systems are using Flash to complement the storage capabilities of HDD. Once Flash memory became a common component of solid-state disks (SSDs),the falling prices and increased densities have made it more cost effective for many other applications. Memory devices and most SSDs that use Flash memory are likely to serve very different markets and purposes. Each has a number of different attributes which are optimized and adjusted to best meet the needs of particular users. Research is moving along the following paths for embedded Flash devices:

- scaling down the cell size of device memory,
- lowering voltage operation, and
- increasing the density of state per memory cell by using a multilevel cells

NOR and NAND Flash are the two major Flash types. NOR Flash has lower density but a random-access interface, while NAND Flash has higher density and interface access through a command sequence. Their corresponding structures have already been detailed in Figure 1.6 (Chapter 1). NAND Flash cell size is much smaller than NOR Flash cell size ($4F^2$) compared to ($10F^2$) because NOR Flash cells require a separate metal contact for each cell. A conventional FG memory device must have a tunnel oxide layer thickness of 8 nm to prevent charge loss and to make 10 years' data retention certain. This necessity will limit scalability for Flash memory devices. Thus, in order to meet technology scaling in the field of memory and data storage devices, mainstream transistor-based Flash technologies will be developed gradually to incorporate material and structural innovations.

A traditional FG memory cell consists of a MOSFET that is modified to include polysilicon as a charge storage layer surrounded by an insulated inner gate

(floating gate) and an external gate (control gate). Charge is transferred to or from the floating gate through a thin (8 to 10 nm) oxide [84, 18]. Since, the floating gate is electrically isolated by an oxide layer, any electrons passing by are trapped there. Flash memory works by adding (charging) or removing (discharging) electrons to and from a floating gate. When electrons are present in the floating gate, current cannot flow through the transistor and the bit state is '0'. This is the normal state for a floating gate. When electrons are removed from the floating gate, current is allowed to flow and the bit state is '1'.

The FG memory has achieved high density, good program/erase speed, good reliability, and low operating voltage for Flash memory applications. However, semiconductor Flash memory scaling is far behind CMOS logic device scaling. The relentless scaling device dimensions has introduced many challenges like retention, endurance, reduction in the number of electrons in the FG, dielectric leakage, cell-to-cell cross talk, threshold voltage shift, and reduction in memory window margins.

NAND Flash Read program operations (i.e. writes) take place to a page, which might typically be 4 KB in size, while erase operations take place to a block, which might be 128KB in size. Erasing a block sets all bits to 1 (and all bytes to FFh). Programming is necessary to change erased bits from 1 to 0. The smallest entity that can be programmed is a byte. Some NOR Flash memory can perform READ-While-WRITE operations. Although NAND FLASH cannot perform READs and WRITEs simultaneously, it is possible to accomplish READ/WRITE operations at the system level using a method called shadowing. Most commercially available flash products are guaranteed to withstand around 100,000 write-erase-cycles, before the wear begins to deteriorate the integrity of the storage. Random access time on NOR Flash is specified at 0.075μs; on NAND Flash, random access time for the first byte only is slower $\sim$ 25μs. However, after initial access has been made, the remaining 4095 bytes are shifted out of NAND at a mere 0.025μs per byte. Table 2.1 serves as a reference point and highlights the salient features of NAND flash memory and the conventional memories that dominate the market today.

2.2.2 MRAM

Magnetoresistive random access memory (MRAM) is a type of non volatile random access memory technology that has been under development for a few decades now. Due to it's numerous like low switching current, high endurance, non-volatility, radiation hardness and density coupled with multiple flavours that can cater to different levels of the requirements

(performance/energy/density), magnetoresistive memories (encompassing it's multiple variants) could dominate the memory hierarchy.

Conventional MRAM

One of the key breakthroughs that has triggered the massive increment in the density of HDDs over the last two decades was the development of incredibly sensitive sensors for the detection of the weak magnetic fields associated with the data-bearing magnetic transitions on the disk. Work on these sensors carried over to a closely related device, now known as the magnetic tunnel junction (MTJ) [16]. MRAM is basically based on memory cells having two magnetic storage elements (ferro-magnetic plates), one with a fixed magnetic polarity and another with a switchable polarity. These magnetic elements are positioned on top of each other but separated by a thin insulating tunnel barrier to form a MTJ as shown in the cell structure in Figure 2.1. Scientists define a metal as magneto-resistive if it shows a slight change in electrical resistance when placed in a magnetic field. One of the two plates is a permanent magnet set to a particular polarity; the other's field can be changed to match that of an external field to store memory.

The simplest method of reading is accomplished by measuring the electrical resistance of the cell. A particular cell is (typically) selected by powering an associated transistor that switches current from a supply line through the cell to ground. Due to the Tunnel magneto-resistance, the electrical resistance of the cell changes due to the relative orientation of the magnetization in the two plates. By measuring the resulting current, the resistance inside any particular cell can be determined, and from this the magnetization polarity of the writable plate. Typically if the two plates have the same magnetization alignment (low resistance state) this is considered to mean '1', while if the alignment is anti-parallel the resistance will be higher (high resistance state) and this means '0'. Data is written to the cells using a variety of means. In the simplest 'classic' design, each cell lies between a pair of write lines arranged at right angles to each other, parallel to the cell, one above and one below the cell. When current is passed through them, an induced magnetic field is created at the junction, which the writable plate picks up. This approach requires a fairly substantial current to generate the field, however, which makes it less interesting for low-power uses, one of MRAM's primary disadvantages.

MRAM is physically similar to DRAM in makeup, although often does not require a transistor for the write operation. MRAM has been slowly getting off the ground though it is still largely 'in development', and being produced on older non-critical fabs. The only commercial product widely available at this

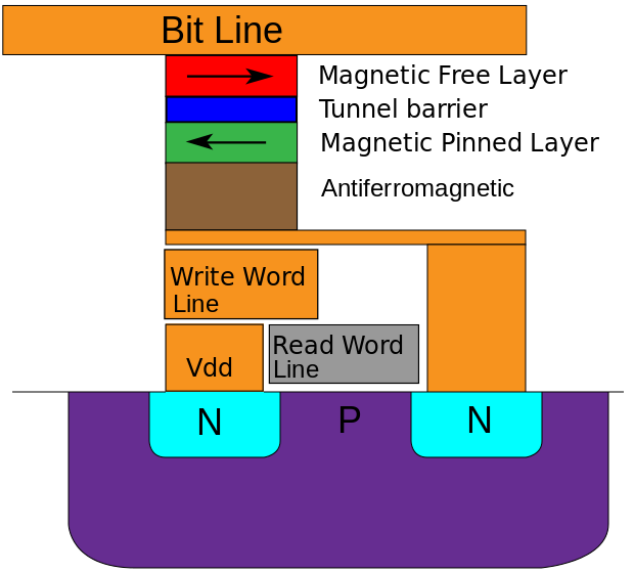


Figure 2.1: Conventional MRAM cell [214]

point is Everspin's 4 Mbit memory [200], produced on a several-generations-old 180 nm process. It has reached some level of commercial success in niche applications [51]. Various companies such as Samsung, IBM, Hitachi and Toshiba, and TSMC are actively developing variant technologies of MRAM chips. In view of power consumption and speed, MRAM competes favorably when compared to other existing memories such as DRAM and Flash, with an access time of a few nanoseconds [143, 202]. Although it has some limitation during the 'write' operation, the smaller cell size could be limited by the spread of the magnetic field into neighboring cells (known as the 'half-select' problem) and needs changes to compete as a universal memory. The price of MRAM is also another issue and considered a limiting factor, with prices far in excess of all the currently established memories at approximately $3 to $5 per megabyte [200]. According to this price level, MRAM is in excess of 1,000 times the price of Flash memory and over 10,000 times the price of hard disk drives. It is expected that of the next-generation memory technologies, MRAM and its variations/derivatives, will cover the biggest market, followed by FeRAM, PCRAM, and memristors. The older MRAM will not considered in this thesis, since it doesn't meet the requirements for our design space exploration (low power and high relative speed).

STT-MRAM

STT-MRAM is a magnetic memory technology that exerts the base platform established by an existing memory called MRAM to enable a scalable nonvolatile memory solution for advanced process nodes [22, 92]. It is a new kind of magnetic RAM with improvements over the traditional MRAM and attractive features like fast read and write times, small cell sizes, potentially even smaller, and compatibility with existing DRAM, SRAM and logic technology. It also has virtually unlimited endurance as compared to other emerging NVM technologies ($\sim 10^{16}$). As we have discussed in the previous section, MRAM stores data according to the magnetization direction of each bit and the nano-scopic magnetic fields set the bits in conventional MRAM. On the other hand, STT-MRAM uses spin-polarized currents, enabling smaller and less energy-consuming bits. The basic cell structure of STT-MRAM is depicted in Figure 2.2. In addition, STT-RAM writing is a technology in which an electric current is polarized by aligning the spin direction of the electrons flowing through a magnetic tunnel junction (MTJ) element. Data writing is performed by using the spin-polarized current to change the magnetic orientation of the information storage layer in the MTJ element [91]. Quite interestingly, the critical current densities required to switch the magnetization in magnetic tunnel junctions are much lower than in metallic pillars, typically by one order of magnitude (in the range of a few 10^6A/cm^2 in MTJ versus a few 10^7A/cm^2 in metallic pillars). This is explained both in terms of higher spin polarization of tunneling electrons in MTJ and angular selection of momentum for the tunneling electrons in MTJ. In a ferro-magnet, spin up and spin down states are separated by a well-defined energy barrier (E_B).The resultant resistance difference of the MTJ element is used for information readout.

STT-RAM is a more well suited technology for future MRAM produced using ultra-fine processes and can be efficiently embedded in subsequent generations of such semiconductor devices as FPGAs (Field Programmable Gate Arrays), microprocessors, microcontrollers, and SoC. A special bonus for embedded designers is the fact that the internal voltage STT-RAM requires can scale down along with the V_{dd} of advanced technology nodes. The difference between STT-MRAM and a conventional MRAM is only in the writing operation mechanism; the read system is the same. The memory cell of STT-MRAM is composed of a transistor, an MTJ, a word line (WL), a bit line (BL), and a source line (SL) [68]. STT-RAM developments have progressed a lot during these last years. Hitachi has produced a 1.8 V, 2 Mb demonstrator chip with a 0.2 μm CMOS technology and MgO-based MTJ bits [90]. In this chip they could demonstrate proper switching with 100 ns write time and 200 μA write current. The read access was 40 ns and the read voltage was optimized as described previously to avoid any disturbance. This chip has been submitted to 10^9 cycles without

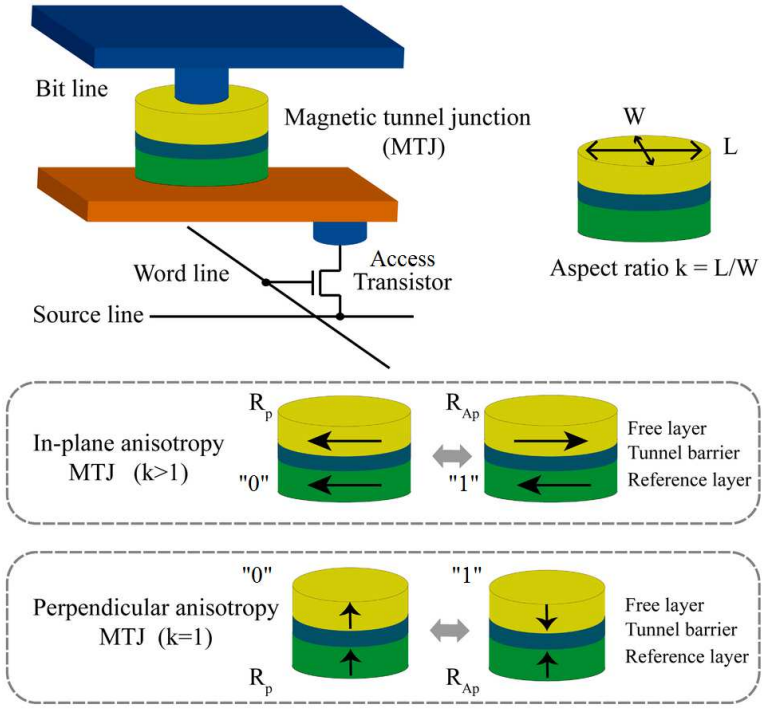


Figure 2.2: STT-MRAM cell and it's operation [4]

degradation of the two states resistance.

Recently, it has been observed that MTJ devices that show a magnetic anisotropy normal to the film surface hold great promise towards faster and smaller magnetic bits in data-storage applications. These perpendicular MTJs have many advantages compared to in-plane magnetized MTJs like low switching current density and reduced cell area (circular instead of elongated) [29]. In-fact the operation current of STT-MRAM composed of 17-nm MTJs has been reduced to 40μA, which is acceptable for memories with medium-scale integration [97].

Currently, STT-RAM is being developed in companies including Everspin, Grandis, Hynix, IBM, Samsung, TDK, and Toshiba. However, for STT-RAM to be adopted as a universal mainstream semiconductor memory, some key challenges should be resolved: the simultaneous achievement of low switching current and high thermal stability. It must be dense (approximately 10-40F^2), fast (below 10 ns of read and write speeds), and operating at low power [5].

From table 2.2, it can inferred that STT-MRAM has met these requirements of late. STT-MRAM still has slightly larger active power for the write operation and/or large write latency when compared to SRAM. Although, write latency can be decreased by improving the MTJ characteristics, read speed cannot be easily decreased since the MTJ has a fundamentally small on-off resistance ratio (still around 150-200% at best). STT-MRAM is expected to compete for the SRAM market either independently (in ultra low power and low speed domain space)or in heterogeneous architectures from N7 onwards. At present drop-in replacements can only be envisioned for LLCs. The explorations presented in chapter 5, 6 and 7 are based on STT-MRAM.

SOT-MRAM

Despite all the advantages of STT-MRAM compared to other NVMs in the market, it does have a few drawbacks. For instance, the write current is still relatively high, leading to a high energy consumption (around 5-10x more energy per write operation than SRAM for in-plane MTJ or even perpendicular MTJ of large sizes) [197][44]. This current has been greatly reduced in recent years with the development of advanced perpendicular MTJs of low pillar dimensions (50nm and below) [95]. IBM has demonstrated switching at 7μm for a 11nm MTJ pillar and a pulse-wdth of 250ns [150]. However, since the current still physically passes through the MTJ, it places a stress on the memory cell and leads to a time dependent degradation of MTJ performance parameters like the TMR, write current, write latency and the lifetime (since the MTJ oxide is threatened by time dependent dielectric breakdown [221] [155]). The write path in itself is also a challenge in STT-MRAM. Since, both the read and write operations share the same access path (through the junction) in STT-MRAM, this can potentially lead to read disturb, i.e., a read operation can by mistake lead to a bit flip (magnetization of the storage layer is switched) [176]. The long write latencies usually prohibit the use of STT-MRAM in first level caches without any micro-architectural modifications to account for the STT-MRAM challenges.

To overcome these issues, Spin Orbit Torque MRAM (SOTMRAM) has been proposed recently ([37] [52]). Spin Orbit Torque (SOT), is a generic term that encompasses different possible mechanisms originating from the bulk of the material and the interfaces between the different layers. These mechanisms issued from the spin orbit interaction are often named the Rashba effect and the Spin Hall Effect (SHE) ([137], [136] and [125]) for the contribution of respectively the interfaces and the bulk. SOT-MRAM uses a three terminal MTJ-based concept to isolate the read and the write path compared to the two terminal concept of STT-MRAM. As a result, in SOT-MRAM the read

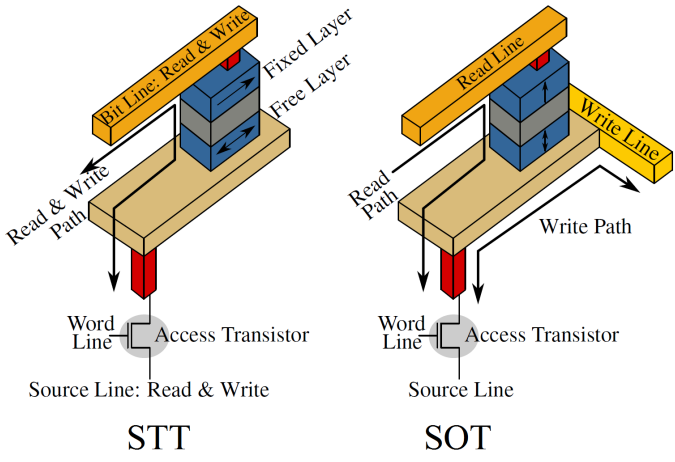


Figure 2.3: STT-MRAM cell vs SOT-MRAM cell [152]

and the write path are perpendicular to each other which improves the read stability [37] [80]. Moreover, the write current is much lower and also the write access is supposed to be much faster, as the write path can now be optimized independently Figure 2.3. Table 2.3 compares SOT-MRAM with other emerging NVMs. In conclusion, it can be said that SOT-MRAM goes a step further when compared to STT-MRAM and can replace SRAM for L1 and L2 levels (Tables 2.1, 2.2 and 2.3) in the near future(N5 node onwards).

2.2.3 RRAM/ReRAM

RRAM is a disruptive technology that can revolutionize the performance of products in many areas, from consumer electronics and personal computers to automotive, medical, military, and space. RRAM is an attractive option despite it's low endurance (since it's disruptive) because it is very compatible with the conventional semiconductor fabrication processes. It also has a high and tunable R-Ratio. Due to these reasons the memristor-based RRAM has been tagged as one of the most promising emerging memory technologies with the potential of being the universal memory by it's proponents [196]. It offers the potential for a cheap, simple memory that could compete across the whole spectrum of digital memories, from low-cost, low-performance applications up to universal memories capable of replacing many of the current market-leading technologies, such as hard disk drives, random-access memories, and Flash memories [217]. RRAM is a simple, two-terminal metal-insulator- metal (MIM)

bi-stable device as shown in Figure 2.4. It can exist in two distinct conductivity states, with each state being induced by applying different voltages across the device terminals.

RRAM uses materials that can be switched between two or more distinct resistance states and researchers have said that electric fields produce conducting filaments through the insulating oxide. Lee [115] has explained that during the SET process, the current level increases from HRS (High Resistance State) to LRS (Low Resistance State) as the voltage increases from 0V to the critical point which is called the set voltage (V set), while the current level abruptly decreases from LRS to HRS at the reset voltage (V reset) under the RESET process. The SET and RESET processes are repeatedly carried out by sweeping the gate voltage with the binary states LRS and HRS. Once the filament is formed, it may be reset (broken, resulting in high resistance) or set (re-formed, resulting in lower resistance) by means of applying voltage above a certain threshold (Figure 2.5).

Oxides have been widely studied for development of resistive memories. The major advantage to using a binary oxide system is the simplicity of the device structure and compatibility with conventional CMOS processing. Some of the RRAM oxides that are the farthest along in development are HfO_x, Cu_xO, NiO, TiO_x and ZrO_x. Important memory attributes of thin Cu_xO have been explored in a 64 - Kb memory array at the 180 nm technology node. The switching mechanism is believed to be due to modulation of the space charge limited conduction through occupancy (vacancy) of deep traps [21]. These devices exhibit very fast switching speeds (50 ns) and low program current (down to 10μA) but show very poor endurance (600 cycles) and insufficient retention.

Memory cells using the TiO_2 material system have been widely studied with a number of different top electrode and bottom electrode materials [50, 28]. The application of an electric field to ionic TiO_2 crystals pulls the oxygen ions away from the crystal toward the top electrode, creating oxygen vacancies that form the conductive path in the ON state [28]. For thick TiO_2 (20 nm) [28], the conductive path density is extremely low, raising questions about scalability. For the HfO_2 system, although low switching voltages ($\sim$ 1.5V) and currents ($\sim \mu$A) have been observed, endurance is very limited and a challenge ($\sim 10^6$ - 10^8 cycles).

Based on it's switching behaviour, ReRAM can be classified into two different types: Unipolar and Bipolar. Unipolar and Bipolar switching behaviour are differentiated by the polarity of the applied voltage. RRAM can also can be classified into four parts with respect to the resistive switching mechanism: (1) Anion-based RRAM (popularly known as OxRAM): the mechanism of this

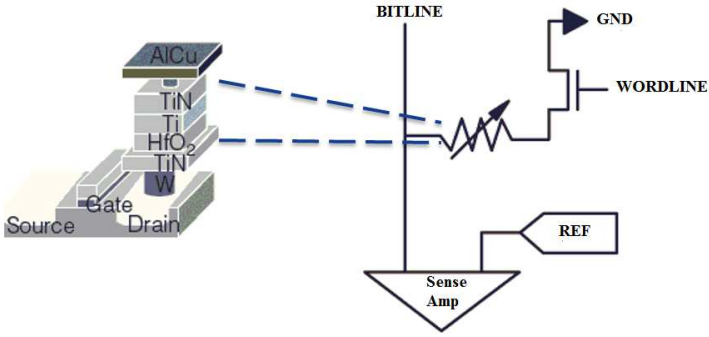


Figure 2.4: RRAM cell and it's operation

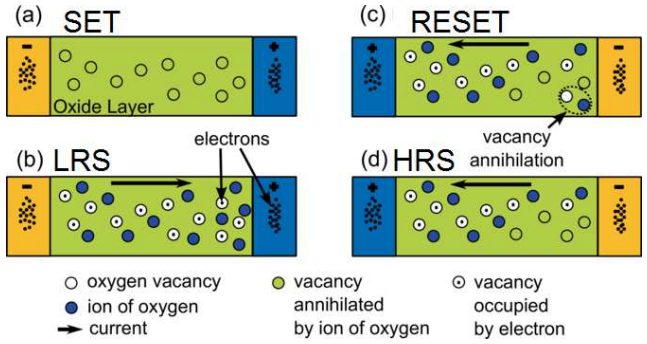


Figure 2.5: An example of the conductive filament formation via Oxygen vacancies. Reproduced from [215]

type of RRAM is dominated by oxygen ions (negative charge), (2) Cation-based RRAM: the mechanism of this type of RRAM is dominated by the redox reaction and migration of metal ions (positive charge) and is usually termed as conductive bridge RAM (CBRAM), (3) Carbon based RRAM: which has carbon-based materials as the RRAM switching layer and (4) Non filamentary RRAM termed as Vacancy Modulated Conductive Oxide (VMCO) RRAM. Tables 2.2 and 2.3 help in comparing these classes of RRAM except for the carbon based RRAM which is comparatively less promising (due to scaling issues).

Anion-based RRAM/OxRAM

Kamiya et al. have concluded by means of a theoretical mechanism that RRAM shows filamentary-type resistive switching, where the oxygen vacancies form conductive filamentary pathways in the resistive material [85]. In-short, it is the formation and disruption of these filaments are the mechanisms responsible for the ON-OFF switching in RRAM devices. The key issue is, therefore, to reveal electronic roles in the formation and disruption of these vacancy filaments. Anion-dominated switching mechanisms and bipolar switching characteristics are usually observed in oxide-based materials, including metal-oxide based and silicon-oxide based materials. Among metal-oxide materials, high-k metal-oxide based RRAMs, such as WO_x, HfO_x, TaO_x and AlO_x have been widely studied due to their attractive performance, along with compatibility with the CMOS fabrication technology, especially in highk/metal gate IC fabrication technology node. Most of the initial work in this PhD thesis (chapter 3 and 4) has been carried out based on the HfOx based OxRAM device presented in [25]. This was suitable for ultra low power domain with a very high read/write ratio. Presently, since it hasn't been possible to scale the switching voltage of these devices (SET V $\sim$ 1.2V, RESET V $\sim$ 1.5V), they are being considered as SCM-M and SCM-S candidates.

Cation-based RRAM/CBRAM

Due to the switching mechanism being dominated by the redox reaction and migration of metal ions, the formation and dissolution of metal filaments lead to the resistive switching behavior of cation-type RRAM (solid-state electrochemical reaction with Ag or Cu to form or dissolve metal filaments) [108]. CBRAM has been widely studied due to its attractive performance, such as very large R_{ON}/R_{OFF}ratio ($> 10^6$), small operating voltages, and also compatibility with the CMOS fabrication technology, especially in Cu-interconnect technology nodes [63]. CBRAM usually has a higher switching voltages when compared to OxRAM, but a better R-Ratio. They can be considered as SCM-M and SCM-S candidates.

Carbon based RRAM

Amorphous carbon (α C) is a non-crystalline carbon allotrope in which the long range crystalline order is not present. This carbon allotrope has been shown to exhibit resistive switching behavior for nonvolatile memory application (induced by hydrogenation and dehydrogenation of H atoms) [19]. A number of switching mechanisms for the have been reported in this case, which

include thermo-chemical sp2 carbon chain forming/rupture, electrochemical metallization and valence change. Memory devices based on α COx are said to exhibit outstanding non-volatile resistive memory performance, such as switching times in the order of 10 ns and cycling endurance in excess of 10^4 times [180].

Vacancy Modulated Conductive Oxide based RRAM

Vacancy Modulated Conductive Oxide (VMCO) is a low switching current resistive memory device in which the conductance of a tunnel barrier is electrically modulated to store memory states. This enables self-rectifying and self-compliant device operation, thus making the integration in 3D stacks simpler by eliminating an additional selector element [60]. The key features of VMCO devices, include: gradual Set/Reset operations, switching currents in the range of a few μA [58], and forming-free operation (since the initial state begins at the low resistive state (LRS)). ON and OFF state resistances show a 1:1 dependence on area, indicative of a non-filamentary switching mechanism. Some examples includes TiO_2 or Al_2O_3 based devices. The TiO_2 cells (in [58]) have a reset switching current density of $0.3MA/cm^2$ (and 10x lower set current). This corresponds to $\sim 5\mu$A reset current in a 40nm-size cell, projecting down to 1μA for a 20nm-size. The cells can be operated with pulses as short as 10ns, at/below +/-7V. Cycling for at least 10^6 cycles and retention of 55 degree centigrade for 3 years are demonstrated. VMCO can be considered as a SCM-S candidate.

RRAM: Concluding notes

Wang and Tseng and Lin et al. have indicated that the interface plays an important role in enhancing the performances of RRAM [212, 121]. Also, recently, Goux et al. have explained that using a stacked RRAM structure has been shown to be one of the most promising methods to improve the memory characteristics [57]. However, critical issues for the future development of RRAM devices remain, such as data retention and memory endurance [182]. A data retention time of over 10 years can be extrapolated from retention characteristics measured at high temperatures and a memory endurance of over 10^6 cycles. Therefore, a statistical study of reliability, availability, and maintainability is essential for the future development of RRAM. HP Labs has planned release prototype chips based on 'memristors' in which migrating oxygen atoms change resistance. Xu et al. have also defined that among all the SCM technology candidates, RRAM is considered to be the most promising as it operates faster than PCRAM and it has a simpler and smaller cell structure

than magnetic memories (e.g., MRAM) [217]. In contrast to a conventional MOS-accessed memory cell, a memristor-based RRAM has the potential of forming a cross-point structure without using access devices, achieving an ultra-high density. This device is based on the bistable resistance state found for almost any oxide material, including NiO, ZrO2, HfO2, SrZrO3, and BaTiO3 [216, 124, 183]. Presently, Samsung, imec and IBM are also actively investigating different variations of RRAM.

2.2.4 PCRAM/PRAM

PCRAM, also known as PCM and chalcogenide RAM (CRAM), is a type of nonvolatile RAM based on a class of material called chalcogenide glasses that can exist in two different phase states (e.g., crystalline and amorphous) [163]. The basic PCRAM cell structure is depicted in Figure 2.6. Most phase-change materials contain at least one element from group 6 of the periodic table, and the choice of available materials can be further widened by doping these materials [2, 174, 175]. In particular, the most promising are the GeSbTe alloys which follow a pseudo binary composition (between GeTe and Sb2Te3), referred to as GST. These materials are in fact commonly used as the data layer in re-writable compact disks and digital versatile disks (CD-RW and DVD-RW) where the change in optical properties is exploited to store data. The structure of the material can change rapidly back and forth between amorphous and crystalline on a microscopic scale. The material has low electrical resistance in the crystalline or ordered phase and high electrical resistance in the amorphous or disordered phase. This allows electrical currents to be switched ON and OFF, representing digital high and low states. This process has been demonstrated to be on the order of a few tens of nanoseconds [9], which potentially makes it compatible with Flash for the read operation, but several orders of magnitude faster for the write cycle. This enables PCM to function many times faster than conventional Flash memory while using less power. In addition, PCM technology has the potential to provide inexpensive, high-speed, high-density, high-volume nonvolatile storage on an unprecedented scale. The physical structure is three-dimensional, maximizing the number of transistors that can exist in a chip of fixed size.

PCM is sometimes called perfect RAM (PRAM) because data can be overwritten without having to erase it first. It is also called ovonic unified memory (OUM). The maximum current needed to operate PCRAMs is the reset current (crystalline to amorphous change). Because reset current is thought to be proportional to the volume of a PCRAM cell, it is expected to become smaller in accordance with the scaling down of PCRAMs. However, the reset current values so far reported are too large to allow PCRAMs to be applied

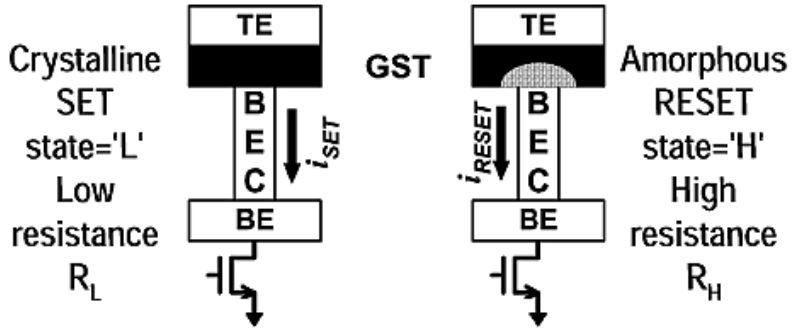


Figure 2.6: PCM cell and it's operation, TE and BE stand for Top and Bottom electrode. The bottom electrode (BE) is in contact of GST material through a narrow bottom electrode contact (BEC). This BEC structure increases the current density of the cell, therefore, the heating efficiency as well. [209]

to the most-scaled LSIs such as 128-Gb flash memories. To reduce the reset current, various types of PCRAM have been proposed, and most of them have been miniaturized beyond the limitation of lithography technology. Reset-current scaling was investigated using devices with microtrench structures recently. In this case, reset current can be reduced below 100μA if the minimum feature size 'F' is reduced to as small as 20 nm. However, the current density exceeds 20MA/cm^2 at the same time. Such a high current density is not acceptable for wires in LSIs. Working prototypes of PCM chips have been tested by IBM, Infineon, Samsung, Macronix, and others. Also, the production of PCM has been announced by in a venture between Intel and STMicroelectronics as well as Samsung. PCM is better suited as a SCM-M candidate (Table 2.2).

2.2.5 FeRAM

FeRAM is a non-volatile RAM that combines the fast read and write access of DRAM cells, consisting of a capacitor and transistor structure as shown in Figure 2.7. The cell is selected via the accessed transistor, which enables the ferro-electric state of the capacitor dielectric (between two electrodes) to be sensed. The polarization properties of a ferro-electric substance enables it's use in a memory device. Today's FeRAM uses lead zirconate titanate (PZT); other materials are being considered. The main developer of FeRAM

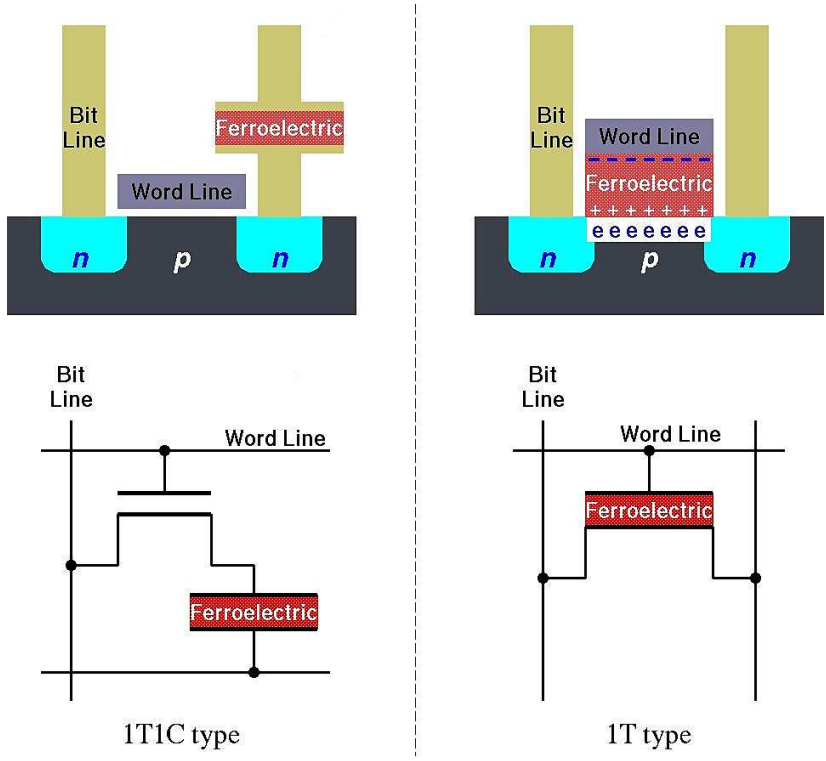


Figure 2.7: FeRAM cell for the 1T and 1T-1C options. [145]

is Ramtron International (acquired by Cypress Semiconductor in 2012). The most straightforward way to detect the state of such a ferro-electric capacitor is to apply a voltage pulse to take the device to one extreme of its hysteresis loop, producing a current spike whose magnitude depends on the initial state. This resulting electric field polarizes the ferro-electric capacitor which disappears as soon as the electric field disappears. The readout voltage produced by the charging of the bitline capacitance by this current can be compared with a reference voltage[188]. However, this readout technique is destructive because the device ends up in the same final state independent of the original data. Thus, each read access must be accompanied by a subsequent rewrite operation, which immediately pushes the required switching-cycle endurance to a very large number. For instance, even at a clock speed of 100 MHz, the worst-case scenario over a 10-year lifetime represents more than 10^{16} read cycles.

Despite this issue, ferroelectric RAM (FeRAM) was one of the strongest early

candidates to be the next nonvolatile RAM because of its inherent speed (as low as 20ns [49]), its low-power, and low-voltage operation [89]. Initially, FeRAM was seen as a way to simply have all the advantages of DRAM together with long-term nonvolatile storage. However, despite years of concentrated integration efforts, FeRAM cells remain larger ($20-40F^2$) than the DRAM cell size of $6-8F^2$[96]. As with any well-explored technology, the problems are now well known. Since the output signal depends on transferring a charge onto the bitline capacitance to obtain a detectable voltage difference, the scaling of FeRAM to smaller device areas with smaller capacitor area inherently leads to smaller signal levels. Thus, the fabrication of FeRAM devices has moved from strapped devices, where the capacitor sits next to the wordline, to stacked devices, where the capacitor sits directly above the wordline, to 3D devices where the capacitor is deposited in a smooth continuous layer either within a trench or over a ridge, augmenting the effective capacitor area without increasing the device footprint [96].

The ferroelectric capacitors in FeRAM devices also tend to show significant problems that include fatigue (the remnant polarization decreases with cycling), imprint (a device left in one state tends to favor this polarization, causing the hysteresis loop to shift), and retention (loss of stored polarization over time) [96, 185, 39]. Ferro-electric memories are expected to have many applications in small consumer devices such as personal digital assistants (PDAs), handheld phones, power meters, and smart cards, and in security systems. It is also expected to replace EEPROM and SRAM for some niche applications and to become a key component in future wireless products. Although, FeRAM has achieved a level of commercial success, with the first devices released in 1993 [139], and current FeRAM chips offer performance that is either comparable to or exceeding current Flash memories [34, 33], integration problems have prevented it from being used widely as a SCM. FeRAM device characteristics are compared to other mature NVMs in table 2.2.

2.2.6 DWM

Racetrack

Conventional Domain Wall Memory or **Racetrack memory** uses spin-polarized electric current to move magnetic domains along a nanoscopic permalloy 'U' shaped wire as a pattern of magnetic regions with different polarities. The U-shaped magnetic nanowire is an array of keys, which are arranged vertically like trees in a forest as shown in Figure 2.9. Achieving capacities comparable to hard drives would require stacks of these arrays. The nanowires have regions with different magnetic polarities (or magnetic

domains), and the boundaries between the regions represent 1 or 0s, depending on the polarities of the regions on either side [75, 159]. The magnetic information itself is then pushed along the wire, past the write and read heads by applying voltage pulses to the wire ends. The magnetic pattern to speed along the nanowire, while applying a spin-polarized current, causes the data to be moved in either direction, depending on the direction of the current. A separate nanowire perpendicular to the U-shaped 'racetrack' writes data by changing the polarity of the magnetic regions. A second device at the base of the track reads the data. Data can be written and read in less than a nanosecond.

Such a memory using hundreds of millions of nanowires would have the potential to store vast amounts of data [160, 158]. In this way, the memory requires no mechanical moving of parts and it has a greater reliability and higher performance than HDDs, with theoretical nanosecond operating speeds. For a device configuration where data storage wires are fabricated in rows on the substrate, conventional manufacturing techniques are adequate. However, for the maximum possible memory density, the storage wires are proposed to be configured rising from the substrate in a 'U' shape, giving rise to a 3-D forest of nanowires. While this layout does allow high data storage densities, it also has the disadvantage of complex fabrication methods, with so far, only 3-bit operation of the devices demonstrated [160]. As the access time of the data is also dependent on the position of the data on the wire, these would also be performance losses if long wires are used to increase the storage density further. The improvements in magnetic detection capabilities, based on the development of spintronic magnetoresistive sensors, have allowed the use of much smaller magnetic domains to provide higher bit densities. The speed of operation of the devices has also been an issue during development, with much slower movement of the magnetic domains than originally predicted. This has been attributed to crystal imperfections in the permalloy wire, which inhibit the movement of the magnetic domains. By eliminating these imperfections, a data movement speed of 110 m/s has been demonstrated [160].

Another difficulty for racetrack technology arises from the need for high current density ($>10^8 A/cm^2$); a 30 nm x 100 nm cross-section would require >3 mA. The resulting power draw would be higher than, for example, spin-torque transfer memory or flash memory. One limitation of the early experimental devices was that the magnetic domains could be pushed only slowly through the wires, requiring current pulses on the orders of microseconds to move them successfully. This was unexpected, and led to performance equal roughly to that of hard drives, as much as 1000 times slower than predicted. Recent research at the University of Hamburg has traced this problem to microscopic imperfections in the crystal structure of the wires which led to the domains

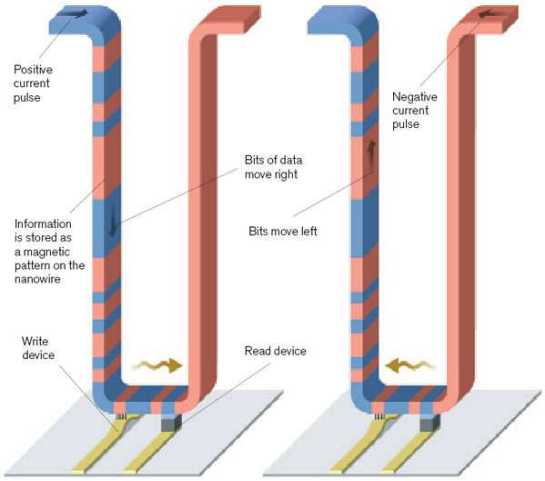


Figure 2.8: Conventional DWM (racetrack) cell read and write programming [160]

becoming 'stuck' at these imperfections. Using an X-ray microscope to directly image the boundaries between the domains, their research found that domain walls would be moved by pulses as short as a few nanoseconds when these imperfections were absent. This corresponds to a macroscopic performance of about 110 m/s[133]. The voltage required to drive the domains along the racetrack would be proportional to the length of the wire. The current density must also be sufficiently high to push the domain walls (as in electro-migration). Presently the lack of progress has limited the prospects of racetrack memory.

TAPESTRI

In [206], the authors propose a new variant of the conventional DWM that is termed as **DWM-TAPESTRI** (Domain Wall Memory TAPE with ShifT based wRIte). The proposed approach, shift based write, offers a fast and energy-efficient alternative to performing writes in spin-based memories. The shift based approach is exemplified by two different bit-cells, TAPESTRI-1bit and TAPESTRI-multi, which are optimized for the differing requirements of the different levels of the cache hierarchy. TAPESTRI-1bit is a bit-cell that is designed to optimize performance. It retains all the benefits of STT-MRAM and can match SRAM in write efficiency. The presence of separate read and write heads helps speeds up the read to the nanosecond regime. TAPESTRI-

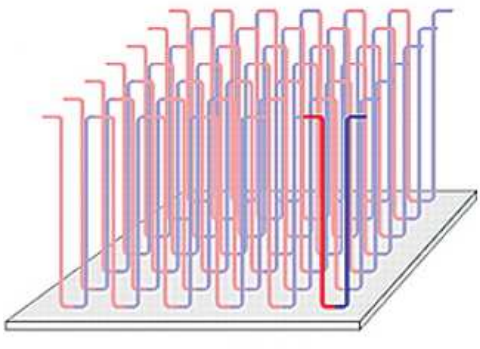


Figure 2.9: DWM array [160]

multi is a bit-cell that is designed to maximally utilize the density benefits of domain wall memory. It achieves much higher density than STT-MRAM (and TAPESTRI-1bit) by storing multiple bits in a single cell. Table 2.3 compared DWM to other emerging NVMs. These advanced DWM variants in their present state of development can be being considered for on-chip memories (LLCs) with micro-architectural modifications alleviate any performance issues.

2.3 Other new memory technologies and the memory roadmap

Other than the above mentioned mature NVM technologies, researchers have also been looking at other potential candidates to match the ever increasing consumer demands. In addition, with growing demand for high-density digital information storage, NAND Flash memory density has been increased dramatically for the past couple of decades. On the other hand, device dimension scaling to increase memory density is expected to be more and more difficult in a bit-cost scalable manner due to various physical and electrical limitations. As a solution to the problems mentioned, NAND Flash memories having stacked layers are under development [74, 224]. In 3-D memories, the cost can be reduced by building multiple stacked cells in vertical direction without device size scaling. As a breakthrough for the scaling limitations, various 3-D stacked memory architectures are under development and expecting the huge market of 3-D memories in the near future (especially for the 10nm node and below). While some of these newer memory technologies not be

looked at to play the role of the 'Universal Memory', almost all of them would target certain niche domains.

The main drive to develop organic non-volatile memory is currently for applications of thin-film, flexible, or even printed electronics. One needs a technology to tag everything to electronic functionality which can be foreseen in a very large quantity and at a very low cost on substrates such as plastic and paper. Accessible popularization of roll-to-roll memory commercialization is a way to make an encounter interesting and challenging to have charge storage devices of choice for applications with enormous flexibility and strength. Recently, polymer (plastic memory) and organic memory devices have significant consideration because of their simple processes, fast operating speed, and excellent switching ability [16, 120]. One significant advantage polymer memory has over conventional memory designs is that it can be stacked vertically, yielding a three-dimensional (3-D) use of space. This means that in terabyte solid-state devices with extremely low transistor counts such as drives about the size of a matchbox, the data persists even after power is removed. Moreover, simple-structure organic bi-stable memory exhibiting superior memory features has been realized by employing various nanoparticles (NPs) blended into a single-layered organic material sandwiched between two metal electrodes [27, 122]. The NPs act as traps that can be charged and discharged by suitable voltage pulses. NP blends show promising data retention times, switching speed, and cycling endurance, but the on-state current is too low to permit scaling to nanometer dimensions [16, 199].

One such class of flexible and organic memory that is being looked into are molecular memories. In molecular memories, a molecular species is the data storage element, rather than, e.g., circuits, magnetics, inorganic materials, or physical shapes [191]. In a molecular memory, a monolayer of molecules is sandwiched between a cross-point array of top and bottom electrodes. The molecules are packed in a highly ordered way, with one end of the molecule electrically connected to the bottom electrode and the other end of the molecule connected to the top electrode, and this molecular component is described as a molecular switch. The Molecular Nanowire memory is an extension of this concept. NRAM is a carbon nanotube (CNT)-based memory, which works on a nano-mechanical principle, rather than a change in material properties [87]. NRAM uses carbon nanotubes for the bit cells, and the 0 or 1 is determined by the tube's physical state: up with high resistance, or down and grounded. NRAM is expected to be faster and denser than DRAM and also very scalable, able to handle 5-nm bit cells whenever CMOS fabrication advances to that level. It is also very stable in its 0 or 1 state. In recent years, DNA has also been shown to be a promising optical material with the material processing fully compatible with conventional polymer for thin-film opto-electronic applications [65, 223].

Researchers from National Tsing Hua University in Taiwan and the Karlsruhe Institute of Technology in Germany have created a DNA-based memory device, that is, write-once-read-many (WORM) times, that utilizes UV light to encode information [76].

Mechanical memories like the 'Millipede memory' and other cantilever based NEMs memories are another class of alternatives being investigated to alleviate the limitations of conventional memories and bring forward certain unique advantages. The millipede memory is non-volatile and stored on nanoscopic pits burned into the surface of a thin polymer layer. It is read and written by a MEMS-based probe. It was developed with the aim to combine the cell area efficiency of hard drives along with DRAM speed. Like a hard drive, millipede stores data in a substrate or medium and accesses the data by moving the medium under the head as well. However, millipede uses many nanoscopic heads that can read and write in parallel, thereby increasing the throughput. Additionally, millipede's physical medium stores a bit in a small area, leading to high storage densities. It was announced in 2002 by IBM [208]. [55] presents a more recent cantilever-based nanoelectromechanical (NEM) nonvolatile memory with a novel write scheme for reliable memory operation at very high-operating temperature (up to 300 degree Centigrade) in rugged electronics.

A lot of these great ideas tend to die before reaching this point of development, but that is not to say that we will be seeing plastic memory on store shelves next year. There are still many hurdles to get over; software alone is a big task, as is the manufacturing process, but active research does bring this technology one step closer to reality. It is not an exaggeration to say that the equivalent of 400,000 CDs, 60,000 DVDs, or 126 years of MPG music may be stored on a polymer memory chip the size of a credit card. Furthermore, the 'functional scaling' trend hints at the emerging NVMs to shift from a silicon base to flexible and transparent redox-based resistive switching memory technologies. Organic-based flexible memories also have merits such as a simple, low-temperature, and low-cost manufacturing process. Several fabrication results of organic resistive memory devices on flexible substrates have been reported [101, 123]. Unlike Flash memory, these new technologies will support in-place updates, avoiding the extra overhead of a translation layer. Also, these new nonvolatile memory devices based on DNA, organic and polymer materials are one of the key devices for the next-generation memory technology where low cost and environmental factors will play a major role in adoption of any new technology. Flexibility is also particularly important for future electronic applications such as wearable electronics. While, today, integration on a silicon chip is not economical, toys, cards, labels, badges, value paper, and medical disposables could be imagined to be equipped with flexible electronics and memory in coming decades. Memory

Table 2.1: Device characteristics of conventional memory technologies ([16] [131] [23] [222] [67]).

Conventional Memories			
Memory Technology	**SRAM**	**DRAM**	**NAND**
Voltage	<1V	<1V	>5V
Read latency	<1ns	∼10ns	10μs
Write latency	<1ns	∼10ns	10μs - 1ms
Write Energy	∼fJ	∼10fJ	∼100pJ
Retention	N/A	∼64ms	>10yrs
Endurance	$>E^{16}$	$>E^{16}$	$>E^{5}$
Cell Area	$<200F^2$	$6F^2$	$<4F^2$

made from tiny islands of semiconductors - known as quantum dots - could also be a potential universal memory candidate, allowing storage that is fast as well as long lasting. Researchers have shown that they can write information into quantum dot memory in just nanoseconds.

From a technology point of view, the monolithic 3D integration of NVM technology and logic circuits via vertical stacking could quite possibly enable a revolutionary digital system architectures of computation immersed in memory. Vertically-stacked layers of logic circuits and memories, with nano-scale inter-layer vias (with the same pitch and dimensions as tight-pitched metal layer vias), provide massive connectivity between the layers. The nano-scale inter-layer vias are orders of magnitude denser than conventional through silicon vias (TSVs). Such digital system architectures can achieve significant performance and energy efficiency benefits compared to today's designs. The massive vertical connectivity makes such architectures particularly attractive for abundant-data applications that impose stringent requirements with respect to low-latency data processing, high-bandwidth data transfer, and energy efficient storage of massive amounts of data. Such concepts are presented in [192, 193]. Presently, 3D NAND is the only memory technology that is feasible and commercially available. 3D NAND is expected to dominate the solid-state drive (SSD) industry from 2017 onward, as suppliers reduce their shipments of flash storage based on traditional 2D or planar NAND.

Tables 2.1, 2.2 and 2.3 highlight the state of the art numbers for the conventional memory technologies, mature NVM technologies and the emerging NVM technologies respectively. These tables provide an insight into the potential of the various technologies to be adopted in the near future and their expected place in the memory hierarchy.

Table 2.2: Device characteristics of mature NVM technologies ([16] [131] [23] [222] [67]).

Mature NVMs				
Memory Technology	**STT-RAM**	**PRAM**	**OxRAM**	**FeRAM**
Voltage	<1V	<3V	<3V	<3V
Read latency	<5ns	∼10ns	∼10ns	20-50ns
Write latency	<10ns	50-100ns	50-100ns	50-100ns
Write Energy	∼0.01pJ	∼10pJ	∼100pJ	∼10pJ
Retention	>10yrs	>10yrs	>10yrs	>10yrs
Endurance	$1E^{15}$	$1E^{9}$	$1E^{6}$	$1E^{12}$
Cell Area	<6-50F^2	6-30F^2	<4-12F^2	20-40F^2

Table 2.3: Device characteristics of emerging NVM technologies ([16] [131] [23] [222] [67]).

Emerging NVMs				
Memory Technology	**SOT-MRAM**	**CBRAM**	**TAPESTRI**	**VMCO**
Voltage	<1V	<3V	<1V	<2V
Read latency	<1ns	∼10ns	1-10ns	10-50ns
Write latency	<1ns	100-200ns	1-10ns	<200ns
Write Energy	∼5fJ	∼100pJ	∼0.001pJ	∼0.1pJ
Retention	>10yrs	>10yrs	>10yrs	>10yrs
Endurance	$1E^{15}$	$>E^{9}$	$>E^{8}$ - E^{12}	$>E^{4}$
Cell Area	<16-80F^2	4-12F^2J	6-100F^2	4-12F^2

2.4 System Architecture Level changes and Hybrid memory systems

2.4.1 Classification of Hybrid Memory Organizations

Going forward, designing systems, memory controllers and memory chips taking advantage of the specific property of non-volatility of emerging technologies seem inevitable. The system architecture co-design aspect is also crucial to take into account the current deficiencies in NVM technologies. This is exceptionally clear from the fact that NVM technologies presently do not match upto their conventional counterparts (from tables 2.1, 2.2 and 2.3). It can be said that emerging technologies enable at least three major system-level opportunities that can improve overall system efficiency:

- Hybrid on-chip cache systems: This level refers to memory levels closest to the processor core and extends from the L0 to LLC, with SRAM used for L1 - L3 and L4/LLC being eDRAM. This is a suitable target for NVMs like STT-MRAM, SOT-MRAM and DWM provided the hybrid cache solutions take into account system architecture design constraints like System Energy, Average Access Time (AAT), and coherency.
- Non-volatile main memory and near main memory space: Here the focus is on the main memory (DRAM presently) and memories close to it (SCM-M). STT-MRAM OxRAM and CBRAM are good candidates for these kind of memories.
- Merging of storage and near storage space: This combines the SCM-S, SSD (Flash) and HDD (Magnetic) layers. PCM, VMCO and CBRAM are good candidates here.

While our work will only focus on the highest level levels of the memory hierarchy and specifically hybrid on-chip caches/scratchpads for system level exploration, we have given a brief outline of the hybrid memory systems at other levels of the memory hierarchy too below. The system level work specific to our explorations will be presented in each chapter.

Hybrid memory systems [20, 42, 161, 198, 70, 129] usually consist of multiple different technologies or multiple different types of the same technology with differing characteristics, e.g., performance, cost, energy, reliability, endurance. An important question that arises here is with regards to data allocation and movement between the different technologies so that one can achieve the best of (or close to the best of) the desired power, performance and endurance metrics [71, 213, 69]. In short, it is best to exploit the advantages of each technology as much as possible while hiding the disadvantages of any technology. Potential technologies include STT-MRAM, OxRAM, CBRAM, PCM and other resistive memories, DRAM, 3D-stacked DRAM, embedded DRAM, PCM, flash memory etc. In short, forms of memory technologies that are optimized for different metrics and purposes, etc. The design space of hybrid memory systems is large, and many potential questions exist. For example, should all memories be part of main memory or should some of them be used as a cache of main memory (or should there be configurability)? What technologies should be software visible? What component of the system should manage data allocation and movement? Should these tasks be done in hardware, software, or collaboratively? At what granularity should data be moved between different memory technologies?

Some of these questions are tackled in [20, 42, 161, 198, 70, 129, 71, 213], among other works recently published in the computer architecture community. These approaches that exploit heterogeneity do increase system complexity but

Table 2.4: Memory array characteristics for the on-chip memory candidates ([45] [131] [23] [222] [67]).

Conventional	NVM alternatives			
Memory Technology	**SRAM**	**STT-MRAM**	**SOT-MRAM**	**DWM**
Level	L1-L3	L2 - LLC	L1-LLC	L1 - LLC
Voltage	<1V	<1V	<1V	<1V
Read latency	<1ns	<5ns	<1ns	<10ns
Write latency	<1ns	10-50ns	10μs - 1ms	100μs
Write Energy	∼pJ	∼10pJ	∼5pJ	∼10pJ
Retention	N/A	>10yrs	>10yrs	>10yrs
Endurance	$>1E^{16}$	$1E^{15}$	$1E^{15}$	$>1E^{15}$
Relative Area	100x	40x	60x	80x

that complexity may be warranted if it is lower than the complexity of scaling SRAM/DRAM chips using the same optimization techniques the industry has been using so far.

2.4.2 Hybrid Memory Systems SOTA

The state of the art for hybrid memory systems for the different (broad) regions of the memory hierarchy are briefly highlighted in the subsections that follow.

NVM hybrid on-chip caches/memory

For the higher levels of the memory hierarchy, the focus is on stringent performance, reliability, endurance and power supply limits with relatively relaxed area. SRAM has been unmatched in performance until now and will be so for the fastest registers that are closest to the processor. However, alternatives such as STT-MRAM, SOT-MRAM, Domain Wall Memory and even some other types of resistive memories with a low switching characteristics (Voltage) like OxRAM can be utilized depending upon the the design space considerations, domain and application. These alternative memories would thrive in ultra low power domains wherein the applications warrant a very high R/W ratio (Reads»»»Writes). Table 2.4 compares some of the more promising alternatives that can serve as alternatives to SRAMs via hybrid solutions. It is quite clear from the table that for the higher levels of the memory hierarchy, there need to be concrete design and system level modifications to match the SRAM metrics.

A number of papers have been published on the the substitution of last level caches by STT-MRAM (advanced perpendicular MTJ) in the near future ([147] [24] and [82] etc). Some have been direct drop-in replacements [147], whereas some have included micro-architectural modifications and design optimizations. In [82], the authors tune the data retention time by trading off the non-volatility and utilize a simple buffering mechanism to ensure the integrity of the application programs given the volatile nature of the tuned STT-RAM cells. The utilization of intermediary structures to reduce writes to the NVM based cache and thus increase the lifetime and performance is a method that has been employed often ([138] and [24]). However, the work on the L1 level has been limited. While there has been some work on hybrid scratchpad memories, these have mostly used STT-MRAM (instead of RRAM) and have a different domain space (general purpose) when compared to our explorations (ultra-low power) [71] [69] [211] [140]. More thorough comparisons of the state of the art with the work carried out in this thesis is presented in chapters 3, 4, 5 and 6.

NVM Main Memory and near main memory space

The non-volatility of main memory opens up a whole new world of opportunities that can be exploited by higher levels of the system stack to improve performance and reliability/consistency (see, for example, [43, 32]). Research on how to adapt applications and system software to utilize fast, byte-addressable non-volatile main memory is an interesting and worthwhile topic to be investigated. On the flip side, the same non-volatility can lead to potentially unforeseen security and privacy issues: critical and private data can persist long after the system is powered down [26], and an attacker can take advantage of this fact. Wear-out issues of emerging technology can also cause attacks that can intentionally degrade memory capacity in the system [184, 170]. Securing non-volatile main memory is this, an important systems challenge. Yoon et al. [220] make the key observation that row buffers are present in both DRAM and PCM (see fig. 2.10), and they have (or can be designed to have) the same latency and bandwidth in both DRAM and PCM. Yet, row buffer misses are much more costly in terms of latency, bandwidth, and energy in PCM than in DRAM. To exploit this, they devise a policy that avoids accessing in PCM data that frequently causes row buffer misses. Hardware or software can dynamically keep track of such data and allocate/cache it in DRAM while keeping data that frequently hits in row buffers in PCM. PCM also has much higher write latency/power than read latency/power: to take this into account, the allocation/caching policy is biased such that pages that are written to more likely stay in DRAM. Note that hybrid memory does not need to consist of completely different underlying technologies.

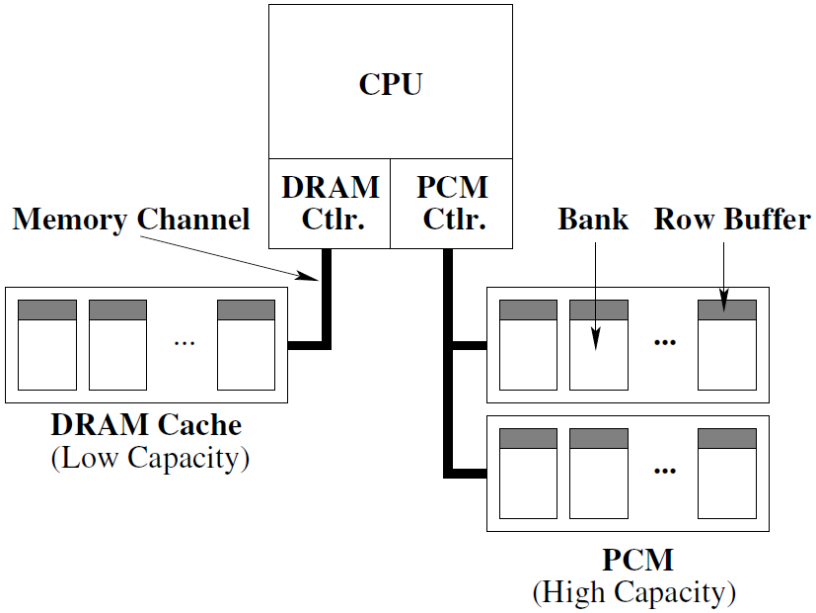


Figure 2.10: An example hybrid main memory system organization using PCM and DRAM chips. Reproduced from [220]

Table 2.5: Memory array characteristics for main memory and near main memory candidates ([45] [131] [23] [222] [67]).

Conventional		NVMs alterntives		
Memory Technology	**DRAM**	**STT-MRAM**	**OxRAM**	**CBRAM**
Level	Main Memory	Main Memory	SCM-M	SCM-M
Voltage	1.2V	<1V	<3V	<5V
Read latency	∼50ns	∼10ns	<50ns	<50ns
Write latency	∼50ns	∼10ns	<200ns	<500ns
Write Energy	∼20pJ	∼20pJ	∼nJ	∼nJ
Retention	N/A	>10yrs	>10yrs	>10yrs
Endurance	$>1E^{16}$	$1E^{15}$	$1E^{15}$	$>1E^{15}$
Relative Area	40x	40x	40x	40x

Table 2.5 compares some of the more promising alternatives that can serve as alternatives to DRAMs and the near main memory space via hybrid solutions. Although we have addressed some of the state of the art with regards to the hybrid nonvolatile main memories and SCM-M memories, its scope lies outside that of this thesis and it will not be covered any further.

Merging storage and near storage space

A promising opportunity that fast, byte-addressable, non-volatile emerging memory technologies opens up is in the design of a systems and applications that can manipulate persistent data directly in memory instead of going through a slow storage interface. This can enable not only much more efficient systems but also new and more robust applications. Traditional computer systems have a two-level storage model: they access and manipulate 1) volatile data in the cache/scratchpad (SRAM today) and main memory (DRAM, today) with a fast load/store interface, 2) persistent data in storage media (flash and hard disks, today) with a slower file system interface. Unfortunately, such a decoupled memory/storage model managed via vastly different techniques (fast, hardware-accelerated memory management units on one hand, and slow operating/file system (OS/FS) software on the other) suffers from large inefficiencies in locating data, moving data, and translating data between the different formats of these two levels of storage that are accessed via two vastly different interfaces, leading to potentially large amounts of wasted work and energy [134]. The two different interfaces arose largely due to the large discrepancy in the access latencies of conventional technologies used to construct volatile memory (DRAM) and persistent storage (hard disks and flash memory).

Today, the new non-volatile memory technologies (NVM), e.g, PCM , OxRAM, CBRAM and VMCO show the promise of storage capacity and endurance similar to or better than flash memory at latencies comparable to that of DRAM. This makes them prime candidates for providing applications a persistent single-level store with a single load/store-like interface to access all system data (including volatile and persistent data). In fact, if we keep the traditional two-level memory/storage model in the presence of these fast NVM devices as part of storage, the operating system and file system code for locating, moving, and translating persistent data from the non-volatile NVM devices to volatile DRAM for manipulation purposes becomes a great bottleneck, causing most of the memory energy consumption and degrading performance by an order of magnitude in some data-intensive workloads[134]. With energy as a key constraint, and in light of modern high density NVM devices, a promising research direction is to unify and coordinate the management of

volatile memory and persistent storage in a single level, to eliminate wasted energy and performance, and to simplify the programming model at the same time.

To this end, Meza et al. [134] describe the vision and research challenges of a persistent memory manager (PMM), a hardware acceleration unit that coordinates and unifies memory/storage management in a single address space that spans potentially multiple different memory technologies (DRAM, NVM, flash) via hardware/software cooperation. There are of-course challenges to be overcome in the design, use, and adoption of such a unit that unifies working memory and persistent storage. These challenges include:

- Coming up with efficient and scalable data mapping, placement, and location mechanisms (that need to be hardware/software cooperative).
- Ensuring consistency and protection requirements of different types of data are adequately, correctly, and reliably satisfied. This also includes the reliable and effective coexistence and manipulation of volatile and persistent data.
- Redesigning applications in such a way that they can take advantage of the unified memory/storage interface and make the best use of it by providing hints for data allocation and placement to the persistent memory manager.
- Providing efficient and high-performance backward compatibility mechanisms for enabling and enhancing existing memory and storage interfaces in a single-level store. These techniques can seamlessly enable applications targeting traditional two-level storage systems to take advantage of the performance and energy-efficiency benefits of systems employing single-level stores. Such techniques are necessary to ensure and a smooth transition to a radically different storage interface.
- Designing system resources in such a way that they can concurrently handle applications/access patterns that manipulate persistent data as well as those that manipulate non-persistent data.

Table 2.6 compares some of the more promising alternatives that can serve as alternatives to flash/storage via hybrid solutions. Akin to the state of the art discussion with regards to the hybrid nonvolatile main memories and SCM-M memories, the scope of SCM-S and emerging NVMs for storage lies outside that of this thesis and will not be covered any further.

Table 2.6: Memory array characteristics for storage and near storage candidates ([45] [131] [23] [222] [67]).

Conventional		NVMs alterntives	
Memory Technology	**FLash**	**PCM**	**CBRAM**
Level	Storage	SCM-S	SCM-S
Voltage	<10V	<3V	<5V
Read latency	10μs	<50ns	<50ns
Write latency	100μs - 1ms	500ns - 1μs	500ns - 1μs
Write Energy	∼10pJ	∼100pJ	∼nJ
Retention	>10yrs	>10yrs	>10yrs
Endurance	$>1E^4$	$1E^9$	$1E^6$
Relative Area	10x	30x	30x

Chapter 3

eNVM based Instruction memory for an Ultra Low Power Platform

3.1 Introduction

Many a times, in reliable real-time systems, all of the assigned task deadlines have to be met under any and all circumstances. As a result of this, it is important to know about an upper bound for the execution times of tasks (WCETs, for Worst-Case Execution Times) [167]. Once, this is known, then the designer of the system can use this with other techniques, such as schedulability analysis to ensure that the system responds fast enough. WCET estimates, in-principle, have to be both safe (greater than any possible execution time) and as tight as possible (as close as possible as the execution time of the longest path - by analyzing the WCET of each path in the program). The ever increasing gap between the processor and the off-chip memory has made the use on-chip memory in real-time embedded systems extremely vital. WCET of programs is quite naturally influenced by the hardware in use. On chip caches have usually been used to bridge this gap. The advantage of caches is that the allocation and de-allocation of memory blocks from the cache are managed by hardware. However, caches can also be a source of predictability problems in real-time systems [166, 66]. Quite a lot of progress has been made with regards to statistical predictions of WCETs of tasks on architectures with caches [141, 66, 119, 127]. Despite this, cache-aware WCET analysis techniques are not always

applicable due to the lack of documentation of hardware manuals concerning the cache replacement policies. Moreover, they also tend to be pessimistic with some cache replacement policies [66, 13]. Finally, caches are sources of timing anomalies in out of order execution processors [128] (a cache miss may in some cases result in a shorter execution time than a hit - a consequence out of resources).

An alternative to caches for on-chip storage is scratchpad memory [165]. Scratchpad memories are small on-chip static RAMs that are mapped onto the address space of the processor at a predefined address range. It can be considered similar to the L1 cache in that it is the next closest memory to the ALU after the processor registers, with explicit instructions to move data to and from main memory, often using DMA-based data transfer. Caches of the same level have uniform memory access latencies when compared to scratchpads with non-uniform memory access latencies. Another difference from a system that employs caches is that a scratchpad commonly does not contain a copy of data that is also stored in the main memory. Scratchpads are not used in mainstream desktop processors where generality is required for legacy software to run from generation to generation, in which the available on-chip memory size may change. They are better implemented in embedded systems, special-purpose processors and game consoles, where chips are often manufactured as MPSoC, and where software is often tuned to one hardware configuration. Their inherent predictability have made them popular in real-time systems. Compared to caches, the task of allocating code/data memory to the scratchpad memory is under software control (it lies with the compiler or programmer). Significant effort has been invested in developing efficient allocation techniques for scratchpad memories [86, 204, 117].

In this chapter, we propose novel eNVM based instruction memory organizations, for such scratchpad based systems, that can address the challenges associated with the write behaviour of the NVMs. We select OxRAM as the NVM of choice for this exploration, since it's characteristics best suit the easy demonstration and implementation of our proposal. These proposals can also be suitably applied to STT-MRAM providing it's characteristics meet certain criteria (like reasonable R-ratio, small read latency and read energy etc). We rethink the SRAM based instruction memory hierarchy for ultra low power embedded systems and replace it by an NVM (OxRAM) based hybrid instruction memory organization. The instruction memory here resembles a Scratch-Pad Memory (SPM) and other higher level software controlled memories. Since the focus is on system level changes, we will not delve deeply into technology and circuit related matters. According to our analysis, even in the worst case scenario, the NVM based hybrid instruction memory organizations showed 87.02% reduction in read energy consumption at 0%

performance penalty.

The major contributions of this work include the following:

- Initially, we analyze the circuit level modifications and optimizations required for the maximal effective use of NVMs (compared to SRAM) with asymmetric read/write (read faster and low energy compared to write). The circuit optimizations help with the transistor sizing, sense amplifier optimization and bit-line loading for the most optimized Energy-Delay product. We then propose a wide word NVM array that can be exploited as such. Here, wide word refers to the bit-width of the word being read/written into the memory. The complete word-line is accessed in our wide word NVM array design.
- We then lay the groundwork for a novel NVM based instruction memory organization that has been micro-architecturally modified for the effective utilization of the NVM and other components. The optimized NVM based L1 wide word instruction memory array is combined with a Very Wide Register (VWR) [173] to produce a unique configuration. The VWR here is re-imagined and utilized as a sort of filter cache [100] (for instructions) instead of a register file (for data). The resulting structure is a novel combination of wide word NVM arrays at both the L0 and L1 levels augmented by VWRs acting as filter caches leading to a highly energy efficient instruction memory organization. The L0, here, is represented by means of a loop buffer. The loop buffer is a small instruction buffer designed to hold frequently executed program loops [205]. Typically, in the first pass of a loop, instructions from the higher levels are fetched and copied to the loop buffer, and these instructions are then used in subsequent passes.
- Finally, along with the basic framework, we also propose several variants to the base NVM based instruction memory organization to tackle different scenarios for the most pareto-optimum output.

This chapter will focus on ultra low power embedded instruction memories for low power wireless/multimedia applications with loop dominated codes. To the best of our knowledge, this is the first such work focusing on a wide array NVM based instruction memory organization, specifically involving a OxRAM based L0 level and the Very Wide Register (VWR) [173], that can also facilitate high throughput if required.

The chapter is organized as follows: Section 3.2 discusses similar work and a recent development with respect to the use of NVMs. Section 3.3.1 lays out certain general OxRAM concepts. Section 3.3.2 outlines the NVM circuit

design and exploration and the NVM array used for our proposed instruction memory organizations is presented. Section 3.5 details the original architecture taken as the base for our test-benches and the NVM based architectures. The various components involved and the motivations for their use are described in detail here. We also analyze write access patterns in the embedded application benchmarks, and discuss the insights that led to the proposed architectures. In section 3.6, the energy modeling and, circuit and technology level considerations are briefly explained. Section 3.7 presents the results of the proposed methodologies, their exploration by means of both real and synthetic benchmarks and its discussion. Section 3.8 concludes the chapter.

3.2 Related Work

Most of the solutions to problems facing hybrid NVM/SRAM proposals for higher level software controlled memories like SPMs include a combination of software (memory mapping, data allocation) and hardware techniques (registers, buffers, circuit level changes). In [129], a PRAM based unified cache architecture for L2 caches on high-end microprocessors is proposed and evaluated in terms of area, performance, and energy. [195] presents a novel approach for redesigning STT-RAM memory cells to reduce the high dynamic energy and slow write latencies. Here, the retention time is brought down by reducing the planar area of the cell, thereby reducing the write current.

In [198], the MRAM L1 cache is supplemented with several small SRAM buffers to mitigate the performance degradation and dynamic energy overhead induced by MRAM write operations. Unlike the other hybrid architectures presented above and the exclusively STT-RAM based architecture presented in [195], our architecture is not based on hardware controlled cache memories. It is primarily a software controlled instruction memory leading to a lower energy and area overhead. More importantly, it is the data allocation to the L0 and it's replacement by a NVM that makes our proposal unlike most of the other work.

Some of the work that is more attuned with our proposals related to low energy software controlled memories in an embedded platform are presented in [70], [140] and [211]. There are two approaches for the mapping of the instruction code (program block) to a software controlled memory: static approach and dynamic approach. In the static approach, a subset of program blocks are transferred to the memory when the application starts and there is no block transfer between the off-chip memory and software controlled memory during the application execution. In the dynamic approach, program blocks can

be transferred between SPM and the off-chip memory during the application execution.

[140] proposes a method, called FTSPM (Fault-Tolerant ScratchPad Memory), which integrates a multi-priority reliability aware mapping algorithm with a hybrid fault-tolerant SPM structure. The proposed structure divides SPM into three parts, and a part is equipped with a NVM which is immune against soft errors. The proposed mapping algorithm is responsible to distribute the program blocks among the three parts with regards to their vulnerability level. The simulation results demonstrate that the FTSPM reduces the SPM vulnerability by about 7x in comparison to a pure SRAM-based SPM. In addition, the dynamic energy consumption of the proposed method is 77% and 47% less than that of a pure NVM-based SPM and a pure SRAM-based SPM, respectively. The main differences between our proposal and [140] lie in the circuit level optimization, including the word access scheme, and the implementation of dynamic transfer of program blocks to the software controlled memory. The circuit level optimization for the SPM array and the wide word access scheme are intrinsic to our proposal and ensure the energy reduction and overall feasibility of our proposal whereas these aspects are not focus of the study in [140]. Additionally, [140] utilizes a new type of commands to the Instruction Set Architecture of processor, named as SPM Mapping Instructions (SMI). SMI commands that stall the processor are executed before the execution of candidate program blocks. After this interrupt, the candidate block is copied from its current address in the off-chip memory to the allocated SPM space, which is registered in the SPM controller unit. Then the execution of the program resumes. In contrast our study utilizes, innate 'loop flags' present in the instruction code for the dynamic allocation of the data.

Hu et al. propose a novel hybrid SPM which consists of non-volatile memory (NVM) and SRAM to take advantage of the ultra-low leakage power consumption and high density of NVM as well as the efficient writes of SRAM [70]. A novel dynamic data allocation algorithm is proposed to make use of the full potential of both NVM and SRAM. According to the experimental results, with the help of the proposed algorithm, the novel hybrid SPM architecture can reduce memory access time by 18.17%, dynamic energy by 24.29%, and leakage power by 37.34% on average compared with a pure SRAM based SPM with the same size area. Similarly, numerous other explorations ([69], [71], [73] and [118]) have been carried out by the same group on the effective exploitation of hybrid scratch pad memories (which consist of a NVM and SRAM) for CMPs by means of novel data allocation techniques and memory configuration. All these above mentioned proposals exploit the ultra-low leakage power consumption and high density of NVM as well as the efficient writes of SRAM. Unlike other existing memory allocation techniques that focus on read/write symmetric

memories, here the algorithm is optimized for read/write asymmetric NVMs. The proposed polynomial-time optimal data allocation algorithm is titled, the CMP Optimal hybRid SPM Data Allocation (CORDA) algorithm [62]. This helps reduce the memory access time and dynamic access energy of the NVM while extending it's lifetime.

[211] explores and evaluates a series of SPM architectures consisting of STT-RAM. The experimental results reveal that with optimized design, STT-RAM is an effective alternative to SRAM for SPMs in low-power embedded systems. [211] is also based on the same dynamic data allocation algorithm (CORDA), highlighted above, that is used to divert the most-written data into SRAM and the most-read data into STT-RAM optimally in a region. The main difference between these proposals for hybrid SPMs and our work, despite working towards the same goal, is the methodology and target. All the proposals mentioned above are focused solely on the data allocation techniques and not micro-architectural changes. They do not utilize a L0 NVM based structure and circuit optimization techniques to mitigate the negative effects of the NVM write access. More-over the focus is on data memory and not instruction memory.

The main focus the work undertaken here is embedded processing platforms running applications with real-time constraints and not High Performance (HPC) cache hierarchy. Another thing to note is that while our focus is on the instruction memory, most proposals are data memory specific. Also, the highest levels that are frequently accessed are still SRAM based in the architecture proposals like [129] and [198], whereas both the frequently accessed higher levels (L0 and L1) are explicitly NVM based in our proposal.

3.3 NVM array design

In this section we detail the technology considerations and assumptions required for the design of an NVM array along with the design considerations and optimizations at the circuit level. OxRAM is taken as the resistive NVM of choice here due to it's maturity (as far as in-house development is concerned also), CMOS compatibility and ease of integration. The terms ReRAM and OxRAM will be used interchangeably in this chapters. It should be mentioned that the design specified here is suitable for resistive NVMs that have a high R-Ratio and small area footprint.

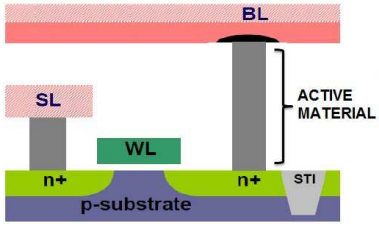


Figure 3.1: OxRAM cell architecture in case of embedded memories

3.3.1 OxRAM: Cell, Characteristics and Parameters

The OxRAM cell taken as reference here is a bipolar Hafnium dioxide cell with Titanium nitride electrodes (X-bar cell of TiN\HfO_2\Hf\TiN from [25]). The 1(T)ransistor - 1(R)esistor model cell stack is used for the embedded OxRAM (Fig. 3.1). The switching is bipolar and we assume relaxed area constraints and minimum mask overhead. The In this set-up, the transistor controls the voltages that are applied to the resistive element (active material). The OxRAM cell dimension is 40nmx40nm and by optimally balancing the SET/RESET pulse conditions, more than 10^{10} pulse endurance cycles can be achieved. The forming voltage is around 2V , the WL voltage on gate of the access transistor ranges from 0.9 V - 1.4 V and the drain voltage via bit-line can range from 1.5 V to 3 V (for RESET, SET voltage is always lesser). We assume that a planar thick oxide access transistor that would be able to drive the required SET and RESET voltage with improvements in OxRAM technology for a pulse-width of about 10ns.

The resistive states of the OxRAM cell are significant since it affects the current drawn for write purposes, the time taken for the Bit-Line (BL) discharge(hence the delay component), and the stability of the cell. In our case, 20 kiloΩ and 1 MegaΩ are the Low Resistive State (LRS) and High Resistive State (HRS) resistance values obtained from cell measurements and calibrations (R Ratio $\sim$ 50). These resistance values for a particular bit-cell design and technology node are not fixed in any way nor optimum for our design considerations. However, these resistance values are consistent with observed values today too [162]. The resistance ratio assumption for design can depend upon the following factors:

- The limitations of the cell itself: whether it can show acceptable variations in resistance values of the two states, the lower limits and the upper limits of the resistance values.
- The technological limitations: drive current for the read and write

operation significantly increases when the resistance values of the cell is decreased. The drive current is proportional to transistor widths. The lower the cell resistance, the larger the transistor required and hence the area of the bit-cell increases. Thus if the transistor sizing is limited by the technology node or other factors it also affects the LRS and HRS values that have to be tailored for the device. The resistance values also determine the pace of discharge across the bit-line and thus the delay across the bit-line.

Similarly, for other resistive NVMs like the STT-MRAM, CBRAM etc the differential resistance ratio is critical. A reasonable R-Ratio ensures fast read access speed and low read access energy. These characteristics are a must for the replacement of SRAM by NVMs at the highest levels (L1 and L0) of the memory hierarchy. An L0 memory is usually very small (2-8KB) and carries out micro-operations. It could comprise of a cluster of registers or be represented by small buffers.

3.3.2 xRAM periphery vs traditional SRAM periphery

The OxRAM periphery and Sense amplifier (SA) model used here are based on the 32 bit SRAM (6T) periphery model for scratch-pad memories used in [35]. The SRAM periphery is suitable for the OxRAM matrix due to similarities like the presence of the access transistor and the high ratio between the current in the LRS and HRS states. However, a number of changes have to be taken into account like:

- RC changes due to technology scaling, different cell stack arrangement and area constraints, including the impact of the less long bit-lines and word-lines (WL).
- The changes due to the cell structure and reduced leakage currents as compared to the SRAM cell (with a CMOS access transistor).
- Single ended sensing: a common reference voltage routed to all the SAs will serve the purpose for the OxRAM context, and no extra energy is consumed.

In this work, the existing drain input latch type voltage-mode sense amplifiers from [181] are used. This is because they have been proven effective in speed and energy performances for small on-chip SRAMs [35]. They do not draw static current, employ positive feedback to provide fast regeneration and their design is rather straightforward.

Bit-line energy and delay is directly related to the bit-line structure. The load capacitance on the bit-line is due to two factors: the capacitance of the bit-line wiring and the drain capacitance of the access transistor. While traditional 6T SRAM matrices use complementary bit-lines, the design in this chapter relies on a single bit-line per cell column. This has several advantages:

- Lesser area is used as compared to double ended lines.
- Lower energy due to minimized area overhead, minimized BL-BL coupling and not using additional select or pre-charge transistors (additional leakage issues).

Taking into account NVM target embedded specifications, namely relatively high speed and low energy, with relaxed constraints on area and low leakage current of non-selected cells, two different bit-line schemes are considered:

- The traditional bit-line scheme in which there is a single bit-line per column that connects to all the cells. Before the read operation starts, this bit-line is pre-charged high via NMOS transistors.
- The divided bit-line scheme [88] where the local bit-lines (LBLs) are selected by means of pass transistors activated by the global word line. The sense amplifiers connect to the global bit lines (GBLs) in this case. In the divided bit-line scheme the effective bit-line capacitance decreases depending upon the number of local partitions. Both the local and global bit-lines are pre-charged high initially and then the pass transistors are activated along with the activation of the word-line. The bit-line discharge across the local bit-line translates to the discharge across the global bit line.

Simulations based on the Energy and delay considerations clearly favour the hierarchical divided bit-line structure over the traditional one. This is because of the reduced capacitive load per bit-line when a cell is being accessed. This in-turn leads a low RC delay and faster discharge propagation along the bit-line to the sense amplifier. Thus, we incorporate the divided bit-line scheme into our design. Keeping in mind OxRAM write performance penalty, one obvious choice is to widen the access granularity. We have thus studied two different access schemes for the OxRAM array structure: the traditional access scheme, with splitting of word lines, and the wide word access scheme. The OxRAM array design with a conventional access scheme closely resembles the structure presented in [35]. However, unlike [35], there is no interleaving of cells.

The wide word access OxRAM array structure is shown in Fig. 3.2. In this structure, there is no splitting of word lines and interleaving of cells. All

Table 3.1: Energy comparisons between the OxRAM and SRAM read access

Parameters	**64kb OxRAM (32bit access)**	**64kb OxRAM (512bit access)**	**64kb SRAM (32bit access)**	**64kb SRAM (512bit access)**
main/wordline Vdd(V)	(0.9/0.9)V	(0.9/0.9)V	(0.9/0.9)V	(0.9/0.9)V
Read energy/ access	0.74pJ	2.20pJ	4.39pJ	32.41pJ
Read energy/ accessed bit	0.023pJ	0.004pJ	0.13pJ	0.063pJ

the cells across the length of the GWL are accessed during a read or write. This reduces the decoder overhead significantly and is the optimal banking implementation to ensure that the advantages of the wide-word access scheme can be exploited. In the OxRAM design used for both the L1-memory and L0 loop buffer, the output is read as a single wide word across the entire length of the word line. Two first-level decoders are used to decode z address bits into two sets of one-hot output wires. The **BLOCK ROW DECODER** (BRD) decodes x address bits into a **BLOCK ROW SELECT** (BRS) signals, and the **WITHIN BLOCK DECODER** (WIBD) decodes the remaining y address bits into b **Within Block Signals** (WIBS). Where,

$$x + y = z,$$

and

$$a * b = 2^z = rows.$$

These within block signals combine with $BRSi$ to generate the word line. All critical timing signals are derived from digitally tune-able delay lines. These delay elements are also used in pulse generators to obtain a tune-able pulse width. All the components are modeled in SPICE and the simulations are carried out via Synopsys HSPICE to predict the timing, power consumption, functionality, and yield of the design. There are no banks/sub-array (along with associated H-tree) required due to small size of the target memory. The netlist here only includes an abstraction of the parasitic capacitances and resistances of the metal lines.

We compare the read access energies for the different access schemes between a 64kb OxRAM (based on our design) with a 64kb SRAM (based on CACTI [142]) (Table 6.1). It is quite clear that, as far as read energy numbers are concerned, OxRAM is more energy efficient as compared to SRAM. This is to be expected. An important deduction from the table is that the wide word access scheme

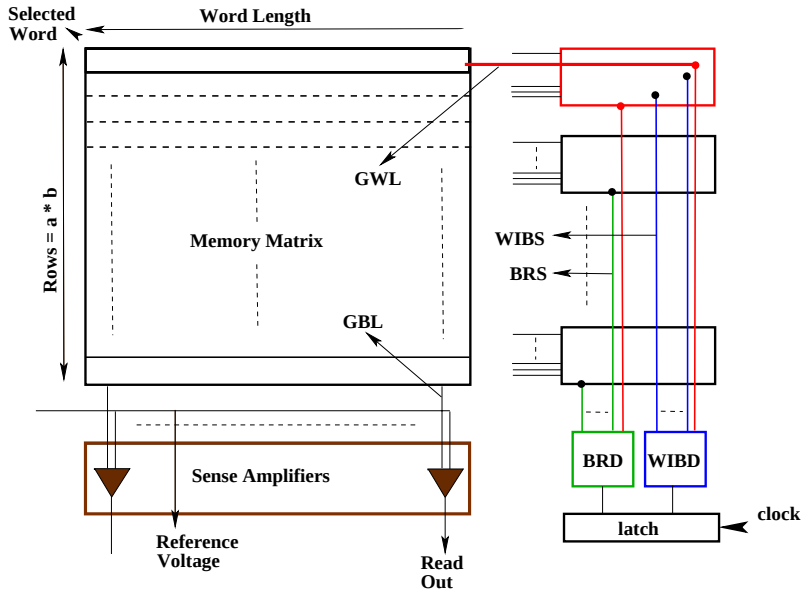


Figure 3.2: Wide word access OxRAM array.

is much more efficient for OxRAM as compared to SRAM. On moving from the traditional access scheme to the wide word access scheme, the energy per accessed bit drops by ∼6 times for the OxRAM configurations whereas it only drops by ∼2 times for the SRAM configurations. This can be attributed to the lack of the cumulative capacitive load of SRAM cells across the entire length of the accessed word line. These energy and delay simulations of the OxRAM periphery clearly indicate that it is feasible and highly desirable to use OxRAM technology in the L1 instruction memory context for deeply scaled embedded process platforms. It has to be mentioned that these numbers were computed at the beginning of the PhD when the OxRAM technology was less mature. This does not match the state of the art data presented in sections 2.3 and 2.4.2. The numbers were based on reasonable OxRAM improvements, voltage scaling assumptions and a planar thick oxide access transistor for the bit-cell derived from the promising data and literature on HfO_2 based OxRAM3.3.1. However, over time it's become clear that OxRAM will require relatively high voltages to switch that demand I/O transistors or a specific selector.

3.4 System Overview

The experimental framework utilized for our explorations emulates the BASE core ([201]). The BASE core is an example processor that is intended to illustrate various concepts related to processor modeling, as well as tool capabilities such as on chip debugging and processor verification. BASE can also be used as a starting point for the development of application specific instruction set processors. BASE contains a collection of general purpose instructions such as:

- 16 bit integer arithmetic, bitwise logical and compare instructions. These instructions are executed on a 16 bit ALU and operate on an 8 field register file.
- Integer multiplications with 16 bit operands and 32 bit results.
- 16 bit shift instructions.
- Load and store instructions from and to a 16 bit data memory with an address space of 64k words, using indirect addressing. Address computations are executed on a 16 bit AGU (Address Generation Unit).
- Various control instructions such a jumps and subroutine call and return.
- Zero overhead loops.
- Support for interrupts.
- Support for on chip debugging.

In addition to these predefined instructions, the user can add application specific instructions to the BASE core. A basic instruction word of the BASE processor is 16 bit wide (data type iword). BASE supports both single word and multi word instructions. Most instructions are single word instructions, where one 16 bit instruction word encodes one or more actions that are performed by the processor. In the case of multi word instructions, the first instruction word also encodes the instruction. The second instruction word extends the first word with a 16 bit immediate value. There is one instruction, doi, which has three instruction words. The second and third instruction word never contain encoding bits. The instruction set is partitioned into 8 groups, based the value of the three most significant bits of the instruction word. The following instruction groups are pre-defined.

- 000 ALU, shift and multiply instructions.

- 001 Register move instructions, and control flow instructions.
- 010 Load and store instructions with indirect addressing.
- 011 Load and store instructions with SP-relative addressing.

Instruction groups with prefix 100, 101, 110 and 111 are available to add extensions to the BASE core. Also the instruction groups listed in the table above are not completely utilized, many combinations within those groups are still available to make extensions.

The system is composed of a data memory hierarchy, an instruction memory organization, a processor, a loop buffer architecture, and an I/O interface (figure 3.3 [8]). In the next paragraphs, the processor and the loop buffer architecture are described in detail. The processor of the system is designed using Target Compiler Technologies (Acquired by Synopsys in 2014). The Instruction-Set Architecture (ISA) of this processor is composed of integer arithmetic, bitwise logical, compare, shift, control, and indirect addressing I/O instructions. Apart from support for interrupts and on chip debugging, this processor supports zero-overhead looping control hardware. This feature allows fast looping over a block of instructions. Once the loop is set using a special instruction, additional instructions are not needed in order to control the loop. This is due to the fact that the loop is executed a pre-specified number of iterations (known at compile time). The special instruction introduces only one delay slot. The status of this dedicated hardware is stored in the following registers:

- **LS** Loop Start address register - It stores the address of the first instruction of the loop.
- **LE** Loop End address register - It stores the address of the last instruction of the loop.
- **LC** Loop Count register - It stores the remaining number of iterations of the loop.
- **LF** Loop Flag register - It keeps track of the hardware loop activity. Its value represents the number of nested loops that are active.

3.5 OxRAM based Instruction Memory Exploration

The original instruction memory organization that is taken as the base for our target applications and test-benches is termed: **Base SRAM Instruction**

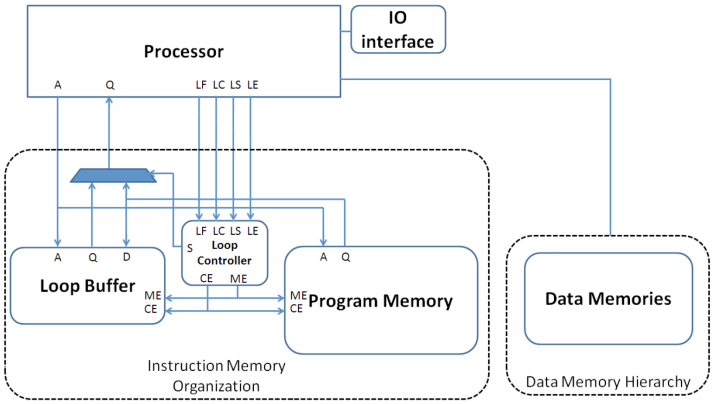


Figure 3.3: Experimental Framework [8].

Memory Organization (BS-IMO). Figure 3.4 is a simplified representation of the same that will used to show our proposed micro-architectural modifications. Here, both the L1 memory and the L0 loop buffer are SRAM based. When executing from the loop buffer the L1 memory can remain idle and switch to a low power state, which has a positive impact on the system power consumption [10] [12]. Unlike the newer variants of loop buffers like the clustered loop buffer organization [81] optimized for low Energy Very Large Instruction Word (VLIW) Embedded Processors and the distributed loop buffer [172] optimized for multi-threading, the loop buffer presented here is a simple variant of the zero overhead loop buffer in [205].

The loop dominated nature of the codes for our target applications leads to more read accesses as compared to write accesses[7]. This read-write asymmetry can be used to exploit the low read energy consumption for OxRAM and alleviate its write access issues like high energy consumption, high delay and low lifetime. The loop dominated nature of the codes also leads to significantly more usage of the L0 loop buffer. This is certainly advantageous considering energy consumption due to the smaller memory size and also from the performance point of view since the L0 here is much closer to the data-path than L1.

3.5.1 Proposed Architectural Changes

The low energy read access of the OxRAM and the application wise read-write asymmetry makes usage of OxRAM alternatives highly preferable for ultra low power embedded systems running the target applications. Other than the

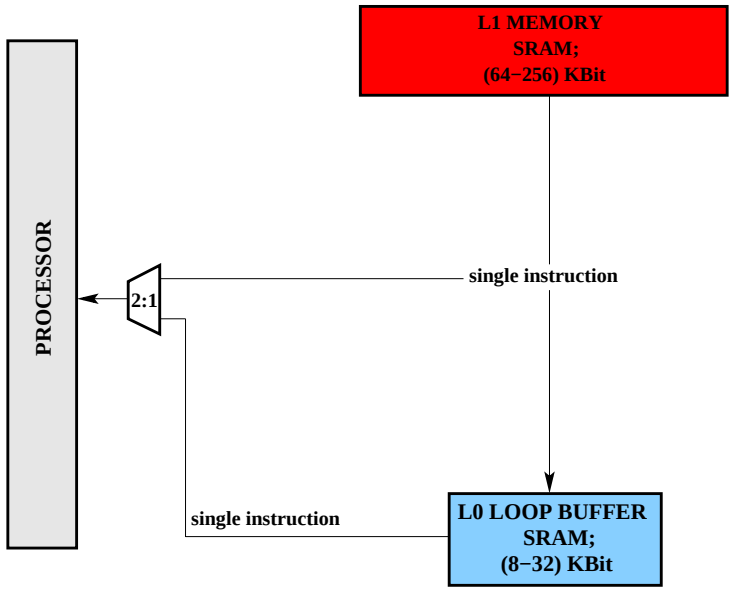


Figure 3.4: SRAM based initial Instruction Memory Organization: BS-IMO.

obvious advantages of non volatility and area, the compatibility of OxRAM with logic technology also makes it suitable to replace SRAM at such high levels (L0).

Thus, we explore several instruction memory organizations envisaging the substitution of SRAM based memory systems by a OxRAM based counterpart. Given the high sensitivity of the OxRAM loop buffer to the increased write latencies, we will carefully explore this L0 layer with specific architectural extensions in order to attain energy savings with no performance loss. First, we propose an architecture to exploit the internal circuit modifications to the OxRAM modules proposed in section 3.3.2 to achieve higher bandwidth at low energy costs. Later, we present different instruction memory organizations as alternatives to meet the timing requirements while saving as much energy as possible.

3.5.2 Addressing the OxRAM limitations

While the read-write asymmetry does alleviate the problems associated with the OxRAM write access to some extent, it is still not enough to make OxRAM a feasible alternative. The write energy consumption will be manageable as a

result of smaller number of write accesses compared to read accesses (read-write asymmetry) and acceptable OxRAM cell write energies compared to SRAM cell write energies [59]. However, the penalty due to the OxRAM write latency will still result in performance penalties. We have assumed an OxRAM write latency of 8 cycles compared to the OxRAM read latency of 1 cycle; a valid assumption as per [59]. One of the ways to limit the write access problems is by the use of wide word access schemes.

Utilizing the wide word access schemes for memories has a number of advantages:

- Writing the wide word into the L0 loop buffer reduces the average write energy consumption per bit.
- Wide word write also reduces the effective total time required to write the frequently executed loops into the loop buffer.
- The wide word read access for OxRAM is much more efficient per bit as compared to that of SRAM because the cumulative capacitive load of the SRAM cells is not present (Section 3.3.2).

The above mentioned reasons are sufficient enough to motivate the transition towards a wide word access scheme for the L0 loop buffer. Reading data from the L1 memory in-case of non loop code will still be highly energy consuming simply due to the size of the memory. Hence, we use a Very Wide Register (VWR) for low energy access of non-loop code and also to facilitate the data transfer from the L1 memory to the L0 loop buffer. A VWR [173] is a register file architecture, which has single ported cells and asymmetric interfaces (Fig. 3.5).

The interface of the VWR, in our case, is wide towards the L1 memory and narrower towards the loop buffer. This asymmetric organization of the register, together with its interface to the wide memory, achieves a significantly higher energy efficiency than conventional organizations. The VWR is always kept as wide as the line size of the background memory (L1 in the current organization), and complete lines are read into it. The VWR has its own multiplexer (MUX), and the controls of the MUX that decide the cell to be accessed can be derived from the program counter itself. The selector lines and program counter help in address generation for the VWR. The VWR helps reduce the performance penalty int two different ways. It functions as a filter cache (thus instructions can be accessed from it) and serving reads with penalty, while also giving high bandwidth output to the L0 loop buffer (thus reducing the total number of writes to L0).

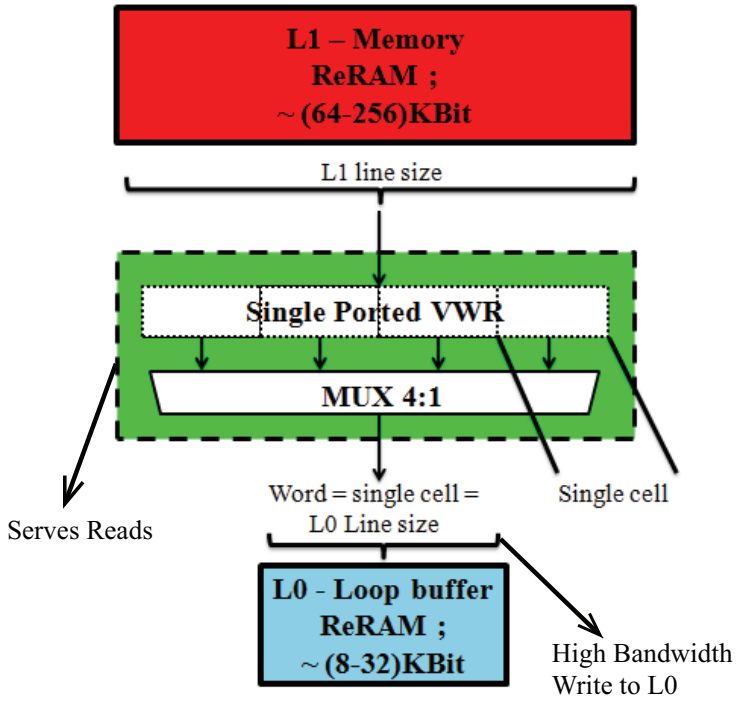


Figure 3.5: Very Wide Register Organization with asymmetric interfaces.

The modified instruction memory architecture; **ReRAM based Modified Instruction Memory Organization** (MR-IMO), is shown in Fig. 3.6. Both the L1 memory and the L0 loop buffer are OxRAM based wide word access memories (32 instruction word access for L1 memory and 8 instruction word access for L0 loop buffer). In the proposed architecture, contrary to [173], we use the VWR as a kind of filter cache (for instructions) rather a register file (for data). Each VWR cell size has a capacity of 8 instructions, which is the word-size/output of the VWR and equal to the line size of the L0 loop buffer. The VWR has a single cycle access similar to register files. We write 8 instructions per write cycle (L0 line size) into loop buffer to minimize the write energy consumption and the performance penalty due to OxRAM write latency.

The L1 line that contains the instruction to be accessed is always transferred to the VWR in case of non-loop codes. Subsequently, they are accessed from the VWR until the program counter encounters an instruction not present in the VWR. The corresponding L1 line is then transferred to the VWR and the cycle continues. The VWR is completely filled in each of its write cycle.

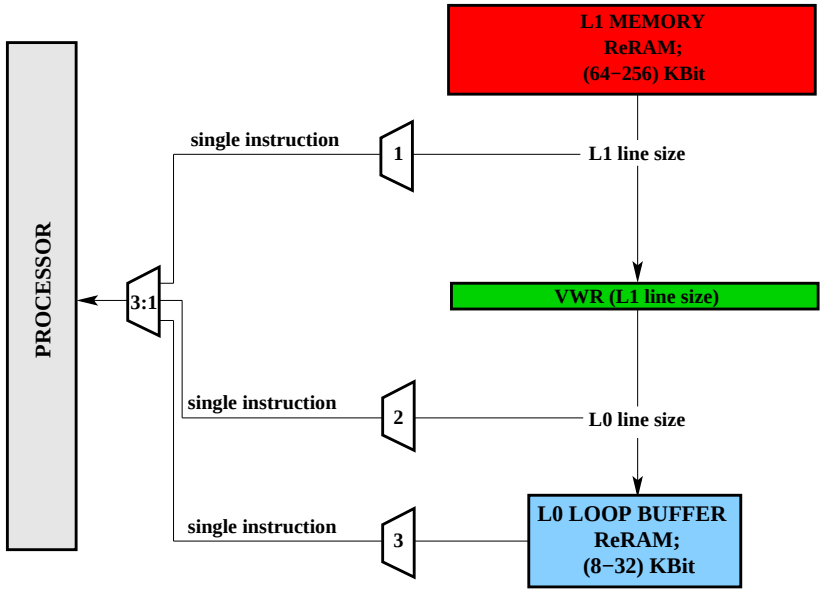


Figure 3.6: Modified hybrid ReRAM based instruction memory organization: MR-IMO.

Multiplexer network 1 extracts a single instruction from wide word written into the VWR towards the processor. This ensures that there is no delay caused by the transfer of data from the L1 to the VWR.

The example given in Fig. 3.7 will help illustrate the data flow in a better manner. The figure is simply a breakdown of the CWT-optimized (Table 3.3) instruction flow. The four main functions (MAIN, QRSDet3C, lmsc and rpeaksearch) that make up the instruction code are highlighted by the blue boxes along with the instruction code sequences they correspond to. The loop bodies are represented by the yellow boxes. The loop body size is indicated beside it (towards the left side) by means of loop 'start' and loop 'end' instructions. The loops are initiated either by the 'do' or the 'doi' instructions. The instruction flow and jumps are specified by means of arrows. The L1 line that contains the instruction to be accessed is always transferred to the VWR in case of non-loop codes in the first pass. Subsequent accesses are from the VWR till the program counter encounters an instruction not present in the VWR. The corresponding L1 line is then transferred to the VWR and the cycle continues. The VWR is completely filled in each of its write cycle.

Multiplexer network 1 extracts a single instruction from wide word written into

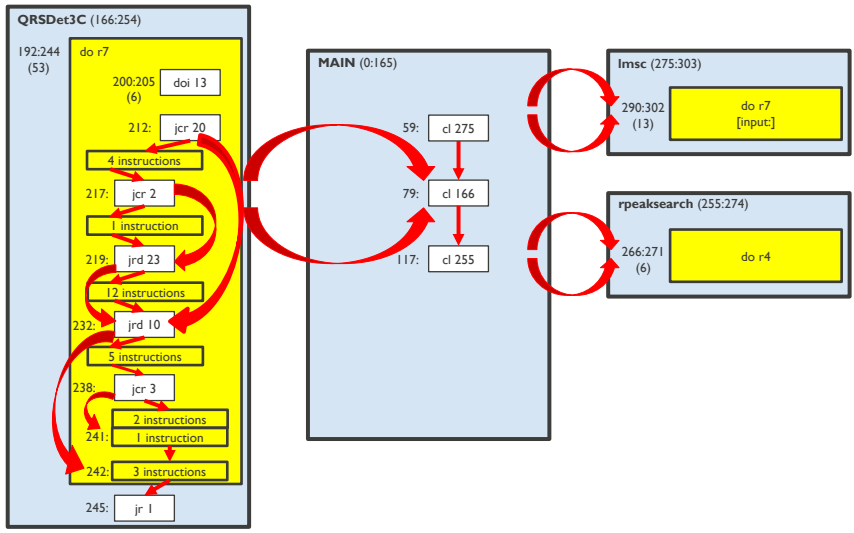


Figure 3.7: CWT-optimized benchmark instruction flow diagram.

the VWR towards the processor. Once the 'do/doi' instruction in encountered, the entire contents of the VWR cell (L0 line size) that contains a loop instruction in question is copied into the loop buffer. Multiplexer network 2 extracts a single instruction from wide word written into the loop buffer from the VWR towards the processor. Multiplexer network 3 extracts a single instruction from wide word read from the loop buffer towards the processor. All the MUX networks are connected to selector lines and the program counter for address generation and appropriate word selection.

As can be seen from Fig. 3.7, there are function calls present within the loop body that call instructions lying outside the loop body. Although, technically the loop flags are still active and it is a part of a particular loop execution sequence, these instructions which lie outside the loop body are not copied into the loop buffer.

The write cycle into the OxRAM loop buffer takes place over 8 cycles as mentioned before. Due to the presence of a smart multiplexer network that selects a single instruction from the 8 instructions being written into the loop buffer, the data can be read from the VWR while it is being written into the loop-buffer. However, the processor has to be stalled in certain situations. These arise when the next instruction to be executed from the code is not present in the VWR cell segment, from which the 8 instructions are being transferred into the OxRAM loop buffer. This leads to a performance penalty.

Due to the OxRAM write latency, instruction codes having short loop bodies and those that are accessed less frequently can lead to significant performance penalty. We can compensate for this in a number of ways. One of the more simpler ways to achieve this would be to read data from the L1 memory directly, when its not present in the VWR cell segment being written into the loop buffer. Since we always read a wide word, this approach can lead to a significant increase in energy consumption.

Another possible option that can be explored is the reading of data from the VWR itself during penalty cycles of the loop buffer write. Since adding an extra read port on the VWR will significantly increase the energy consumption, we introduce a small register in-between the VWR and the loop buffer. This register will be termed as the L0 line register and has the loop buffer line width of 8 words. The data to be transferred to the loop buffer is first transferred to this L0 line register and held for the duration of OxRAM write delay. Meanwhile, data can be read from the VWR independent of the data being written into the loop buffer.

The wide word being read from the L0 loop buffer every time it is accessed consumes more energy than the regular (smaller selective word) access scenario. This is a target for further energy optimization. A smaller VWR (that extracts a single instruction from the L0 line size) is used to exploit the mostly-regular access from the L0 loop buffer and reduce energy consumption even more. This modification is termed as **Energy and Performance Optimized ReRAM Instruction Memory Organization** (EPOR-IMO), (Fig. 3.8).

The presence of 2 different VWR's helps us to exploit locality even more, especially since our target applications have regular loop dominated codes. The energy consumption also drastically decreases when the majority of accesses are to these small energy efficient structures. These proposed modifications do come at the cost of extra hardware. The implementation of 2 different VWR's and a line register also might strain the routing and layout of the organization. However, it must be remembered that the area savings offered by the substitution of SRAM by an NVM helps alleviate these concerns by a significant margin.

We also explore an additional option without a loop buffer in-order to further effectively quantify and deduce the effects of the loop buffer and the VWR on the **Loop Buffer devoid Instruction Memory Organization** (LB-IMO) (Fig. 3.9). In this scenario the performance penalty issue is also non-existent since there is no writing of the data into a OxRAM based loop buffer. While this may appear to be a simple and effective solution for the write latency issue of OxRAM, it can be faced with large energy costs in case of highly irregular code or very large loops wherein the L1 has to be repeatedly accessed as a result

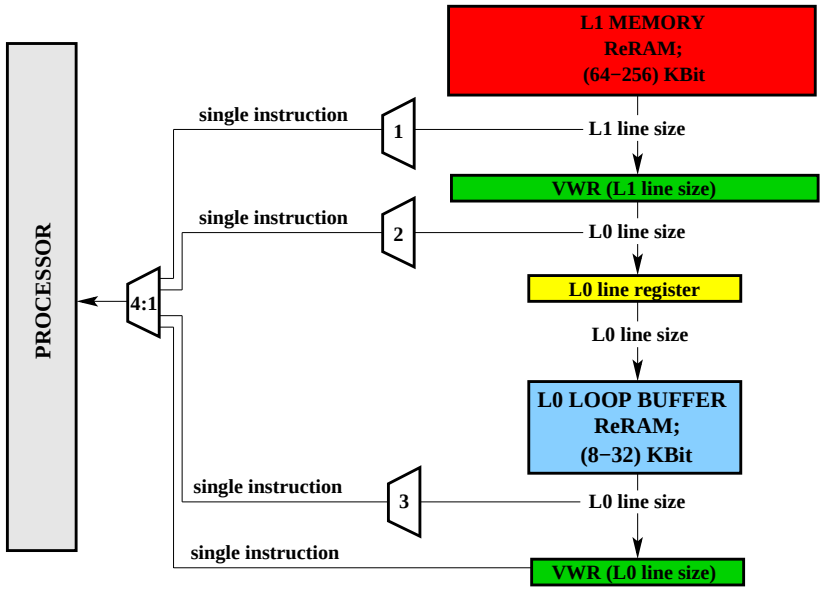


Figure 3.8: Energy and Performance Optimized ReRAM Instruction Memory Organization: EPOR-IMO

of capacity misses. The area and overhead and complexity is vastly reduced in this structure.

3.6 Energy Modeling

The widely used CACTI simulator [142] is extended to calculate the energy SRAM memories in the original instruction memory organization. CACTI is an integrated cache and memory access time, cycle time, area, leakage, and dynamic power model that is purely behavioural in nature. By integrating all these models together, users can have confidence that tradeoffs between time, power, and area are all based on the same assumptions and, hence, are mutually consistent. CACTI is intended for use by computer architects to better understand the performance tradeoffs inherent in memory system organizations. The CACTI cache access model [38] takes in the following major parameters as input: cache capacity, cache block size (also known as cache line size), cache associativity, technology generation, number of ports, and number of independent banks (not sharing address and data lines). As

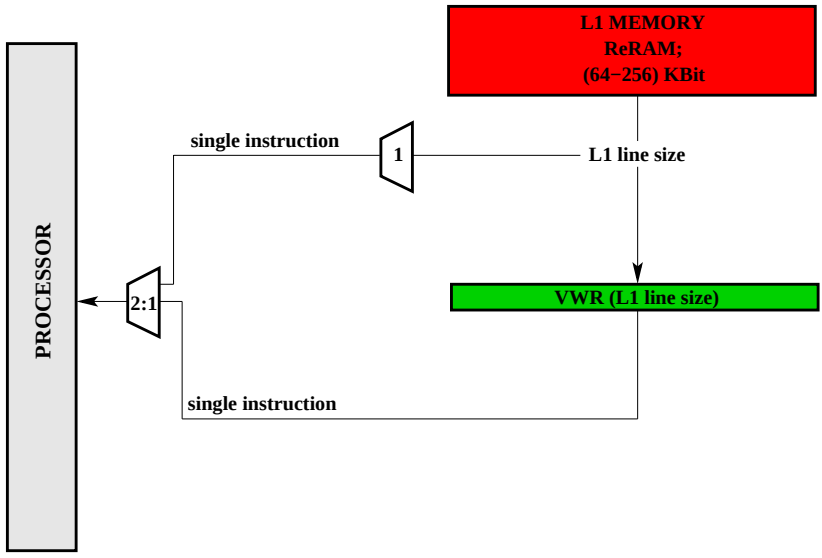


Figure 3.9: Loop Buffer devoid Instruction Memory Organization: LB-IMO.

output, it produces the cache configuration that minimizes delay (with a few exceptions), along with its power and area characteristics.

CACTI models the delay/power/area of eight major cache components: decoder, wordline, bitline, senseamp, comparator, multiplexor, output driver, and inter-bank wires. The wordline and bitline delays are two of the most significant components of the access time. The wordline and bitline delays are quadratic functions of the width and height of each array, respectively. In practice, the tag and data arrays are large enough that it is inefficient to implement them as single large structures. Hence, CACTI partitions each storage array (in the horizontal and vertical dimensions) to produce smaller sub-arrays and reduce wordline and bitline delays. The bitline is partitioned into *Ndbl* different segments, the wordline is partitioned into *Ndwl* segments, and so on. Each sub-array has its own decoder, and some central pre-decoding is now required to route the request to the correct sub-array. CACTI carries out an exhaustive search across different sub-array counts (different values of *Ndbl*, *Ndwl*, etc.) and sub-array aspect ratios to compute the cache organization with optimal total delay. The use of the CACTI behavioural simulator for the energy, delay and area modeling of memories was motivated by its wide adoption in literature, availability and ease of use.

All the memories have in our instruction memory organization have single read

and single write ports. The technology node is 32nm, standard V_{dd} is taken as 0.9V and all the transistors; SRAM cell, global and local circuitry, word line drivers and sense amplifiers etc. are assumed to be ITRS low standby power transistors. All the above mentioned technology considerations are extended for all energy calculations.

The energy numbers and access latencies are calculated based on the PTM cell library (32nm technology node) for the OxRAM based memories and other elements like the VWR and L0 line register. The OxRAM memories have the wide word access array configuration which is detailed in section 3.3.2. The VWR and L0 line registers are D-latch based in our implementations since the timing is entirely regulated by means of a control unit that regulates a state machine. In this implementation, we never write and read from the VWR or the L0 line register simultaneously and hence the D-latch based implementation will suffice instead of the flip-flop based implementation.

The L1 and L0 memories are put into sleep modes when they are not being used. The leakage energy values for the SRAM memories are obtained by means of CACTI. The leakage energy consumed per access for the L1 and L0 OxRAM memories is about 71% and 53% lesser than the SRAM memories respectively for the wide word access configuration (with CMOS access transistors). It is about 65% lesser than the SRAM memories with the normal access configuration for both L1 and L0. The L0 line register has nearly negligible leakage energy. The VWR leakage energy is obtained by means of internal assessment. Once the individual energy contributions have been calculated for active and sleep modes, we use the ModelSim [61] VHDL simulator to obtain the total number and type of accesses of the individual components in the system and by extension the total energy consumption.

Table 3.2 details the energy contributions of the different components used in the various instruction memory organizations. The technology specifications are as mentioned above. The configurations of the different memory elements are given in Section 3.7.

3.7 Exploration and Analysis

All organizations are modeled using VHDL and simulations of complete benchmarks are analyzed real-time via ModelSim. The following assumptions are made in the simulations and tests:

- The L1 memory and L0 loop buffer sizes are assumed to be 64Kbit and 8Kbit respectively. A single instruction (16 bits) is usually read out from

Table 3.2: Dynamic read energy contributions of the different instruction memory organization components.

Component	OxRAM L1	OxRAM L0	SRAM L1	SRAM L0
Read energy per access	2.20pJ	0.57pJ	3.51pJ	1.34pJ
Component	SRAM L1 (wide access)	SRAM L0 (wide access)	VWR 1	VWR 2
Read energy per access	32.41pJ	3.41pJ	0.39pJ	0.28pJ
Component	L0 line register	—	—	—
Read energy per access	0.30pJ	—	—	—

the L1 and L0 in the normal scenario (SRAM based L1 and L0 in BS-IMO). The wide-word L1 memory block is organized as 128 lines of 512 bits wide words. Hence, VWR 1 (the VWR between L1 and L0) capacity is 512 bits. This VWR has 4 cells, each cell corresponding to the L0 line size; 128 bits wide (8 instructions of 16 bits each). The wide word L0 loop buffer is organized as 64 lines of 128 bits wide words. VWR 2 and the L0 line register are 128 bits wide. VWR 2 is organized as 8 cells of 16 bits each. These numbers are not fixed and have been arbitrarily chosen to represent the instruction memory of an ultra low power embedded platform. In such a system, the L1 is assumed to be able to accommodate any kernel to be run before-hand and it will not be subject to dynamic writing during the execution process. This is similar to the ultra-low power DSP platforms like TI TMS320. Thus, these kernels can be loaded before run-time and the delay due to the slow writing into the OxRAM based L1 is avoided.

- Due to the loop dominated codes of our target applications and the wide word access LI-L0, while we require access every cycle for the read mode, the write cycles are greatly reduced. Every loop statement is read at least a factor L times compared to the write access count. The writes are slower than the read with a typical factor K, where the OxRAM memory read latencies can easily be assumed to be the same as that SRAM memory. In practice $L >> K$, where $K \sim$4-8 cycles (based on [190] and proposed read access models), so slow write times are acceptable. From the above and the proposed OxRAM memory structure, we can conclude that modeling the read access is sufficient enough to provide a solid idea about the feasibility of the proposed architectural changes.

- We assume that the data transfer from L1 to VWR, VWR to Loop buffer, VWR to L0 line register and from all components to processor to be single cycle accesses.
- We assume that the size of the loop body of all loops present in the instruction code is lesser than or equal to the loop buffer capacity. All the loop instructions, except for the function calls that lie outside the loop body in question, are transferred into the loop buffer. This is achieved by applying proper code transformations upfront.

3.7.1 Exploration of real Benchmarks

Five different benchmarks are used to test and compare the power consumption and performance of the original and modified instruction memory architectures:

- **DWT**: Implementation of the Discrete Wavelet Transform (DWT) algorithm on an ultra low power DSP; used often for processing and analyzing the electroencephalogram data.
- **CWT base**: Implementation of a robust beat detection algorithm based on Continuous Wavelet Transform Modulus Maxima (CWTMM) of an electrocardiogram into a version that can work in real time on a low power processor within wireless ambulatory cardiac monitor.
- **CWT optimized**: Optimized version of CWT base. The processor core and instruction code is optimized for the application.
- **AES base**: ASIP for data confidentiality, data authentication, data integrity and replay attack protection and the design is appropriate for wireless sensor networks.
- **AES optimized**: Optimized version of AES base. The processor core and instruction code is optimized for the application.

A profile of the instruction codes for the different benchmarks used is given in Table 3.3.

In BS-IMO, all the non loop instructions are read from the larger L1 and not from the VWR like in almost all the proposed OxRAM organizations. Due to this, comparing it's energy consumption to the OxRAM based modified architectures would not be fair (see table 6.1 and section 3.3.2). Thus we have included the **SRAM based Modified Instruction Memory organization** (MS-IMO) in our simulations and analysis. This architecture is similar to

Table 3.3: Instruction code profile of the different benchmarks detailing the static and dynamic instructions along with the number of loops and iterations.

Benchmark	Static inst	Dynamic inst	No. of loops	Iterations
DWT	499	385362	2	6000
CWT-base	627	274464	9	31697
CWT-optimized	303	102827	4	20438
AES-base	1048	707052	17	1584
AES-optimized	913	3347	4	147

Table 3.4: Performance penalty for the different benchmarks.

Benchmark	Performance penalty
DWT	0.063%
CWT-base	0.807%
CWT-optimized	0.29%
AES-base	2.13%
AES-optimized	0.72%

MR-IMO with the OxRAM memories simply being substituted by SRAM. The SRAM memories in MS-IMO are of the wide word access configuration though.

Energy consumption alone cannot motivate the transition from SRAM based instruction memory organizations to ReRAM based instruction memory organizations. Table 3.4 outlines the performance penalties in when the different benchmarks are run on MR-IMO. A simple solution to this problem of performance overhead would be to read directly from the L1 memory during the L0 write cycle. This is an extension of MR-IMO and involves only a slight change to the control unit functioning. EPOR-IMO, however includes changes

Table 3.5: Dynamic read energy contributions of the different instruction memory organization components.

Acronym	Organization
BS-IMO	Base SRAM IMO (Fig. 3.4)
MR-IMO	ReRAM based Modified IMO (Fig. 3.6)
MS-IMO	SRAM based Modified IMO (SRAM variant of MR-IMO)
EPOR-IMO	Energy and Performance Optimized ReRAM IMO (Fig. 3.8)
EPOS-IMO	Energy and Performance Optimized SRAM IMO (SRAM variant of EPOR-IMO)
LB-IMO	Loop Buffer devoid IMO (Fig. 3.9)

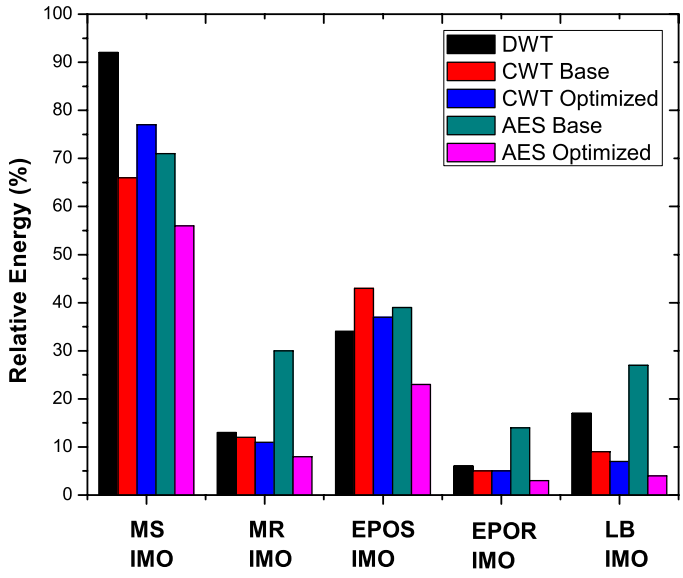


Figure 3.10: Read energy consumptions of the different IMOs based on relative gains. The read energy consumption of BS-IMO across all the benchmarks is taken as the base (i.e. 100%).

to the instruction memory architecture and takes advantage of the regularity of loop access for further energy savings. The read energy consumption for all the proposed architectural modifications and extensions are outlined as a Manhattan chart on the basis of relative gains in Fig. 3.10. The energy consumption of BS-IMO is taken as the base (100%) and the read energy consumption of all other architectures are plotted relative to this. **Energy and Performance optimized SRAM Instruction Memory Organization** (EPOS-IMO) is the SRAM substitute for the most energy efficient ReRAM based IMO; EPOR-IMO. EPOS-IMO however, does not require the L1 line register since writing into SRAM only takes a single cycle. The inclusion of this architecture helps us gauge better the contribution of the individual components and the extent of how energy efficient ReRAM based memories are. Table 3.5 lists the different Instruction Memory Organizations considered here along with their acronyms and also points to their graphical representations (figures).

The SRAM based IMOs like MS-IMO and EPOS-IMO consume the most energy across the different Benchmarks. This is expected since the read access energy of the SRAM based memories is greater than that of the OxRAM

based memories. It is to be noted that the use of the VWR reduces the energy consumption even when the L1 and LO are SRAM based. This can be attributed to the majority of the non-loop code being read from the energy efficient VWR. Looking at EPOS-IMO, there is significantly lesser energy consumption as compared to MS-IMO. This is due to the presence of the last stage VWR. Since, the target applications have loop dominated codes, this VWR is used much more frequently and leads to large energy savings.

We now look closely at the energy consumption across the different benchmarks for both MS-IMO and EPOS-IMO. This will help understand the contributions of the different elements towards the total energy and also predict the applications for wherein our proposals can be most effective. From the instruction profile of the benchmarks (table 3.3), AES-base, DWT and CWT-base are expected to have the largest energy consumption respectively simply going by the number of dynamic instructions. However, we see that it is not so.

In both CWT-base and AES-base, there are a large number of function calls within loops. Since these are not copied into the loop buffer while memory mapping, they are read mostly from the VWR. This is more efficient for the SRAM based IMOs. In DWT there are no function calls at all and the most of the instruction code is a single large loop. Hence, most of the read accesses are from the wide access loop buffer. SRAM memories are not as efficient as OxRAM memories for the wide word access scenario. Thus reading from the wide word SRAM L0 loop buffer leads to a higher energy consumption. CWT-base has the maximum number of function calls within loops across all the benchmarks and thus, the reduced energy consumption. As far as AES-optimized is concerned, it has the least number of dynamic instructions and this leads to its low energy consumption for all proposals, both SRAM and OxRAM.

For the EPOS-IMO proposal, the situation is reversed. Due to the presence of the VWR after the L0 loop buffer, the loop buffer reads are significantly reduced. This leads to huge energy savings and we see that the benchmarks with less function calls (i.e. more 'pure' loops) and higher number of dynamic instructions have lesser energy consumption across the benchmarks (mainly DWT and CWT-optimized to a lesser extent due to lack of function calls).

Even with the reduced energy consumption due the VWR in MS-IMO and EPOS-IMO, energy savings on substituting SRAM memories with OxRAM are visibly more significant. As stated before, this is because the read energy access for the OxRAM is lower and the wide word access for the OxRAM is much more efficient than for SRAM (Section 3.3.2). The leakage energy savings also lead to this relative reduction of energy consumption (Section 3.6). The

leakage energy savings amount for about 9% of the total read energy savings for ReRAM based IMOs such as MR-IMO and EPOR-IMO as compared to MS-IMO and EPOS-IMO. Among the ReRAM based IMOs, it is quite clear that EPOR-IMO and LB-IMO are the most energy efficient; by a factor of 2−3 compared to the other ReRAM based alternatives. EPOR-IMO always has a lower energy consumption as compared to MR-IMO, largely due to the last stage VWR (but at the cost of larger area and complex routing). The reduced energy consumption in LB-IMO can be attributed to both the loop and non loop code being read mostly from the VWR.

The analysis of the energy consumption across the different benchmarks is more or less straightforward in case of the ReRAM based IMOs. It highlights many of the points inferred from the analysis of the SRAM based IMOs earlier in this section. AES-base has the maximum energy consumption across the different benchmarks for both MR-IMO and EPOR-IMO. This is expected since it has the largest number of dynamic instructions across the different benchmarks. These loops are also highly irregular and extend over different lines read from L1 into the L0 via the VWR.

Even with the limited number of writes into the loop buffer, there can be a significant performance penalty for MR-IMO in certain scenarios that have been explained earlier. This aspect has to be thoroughly analyzed. A number of different approaches to reduce the performance penalty have been proposed earlier (Section 3.5.2). Selective mapping of the loops into the loop buffer based on the loop body size and loop iterations can be utilized for the same. The energy and performance trade-off's with respect to loop body size are given in Fig. 3.11 and Fig. 3.12, and with respect to loop iterations are given in Fig. 3.13 and Fig. 3.14.

It is clear from the figures 3.11 and 3.13, and the instruction code profiles (table 3.3) that, in cases of target application codes having a significant number of short loops that are executed less frequently, the performance penalty increases to a considerable amount. If only largest loops with a very large number of iterations are selectively filtered into the loop buffer, the performance penalty decreases drastically. This means that very few loops are written into the loop buffer and the resulting writes (vastly reduced) lead to a negligible performance penalty. However, this filtering leads to an increased number of reads from the larger (thus more energy hungry) L1 and leads to increased energy consumption. Hence, there is a trade-off between energy consumption and performance penalty when we consider the filtering of loops to reduce performance penalty. Moreover, 0% performance penalty can be obtained in this approach by only making the loop buffer void and runs contrary to the aim of reducing energy consumption. Again, it has to be made stressed that it is only the selective mapping/filtering of loops here that leads to performance

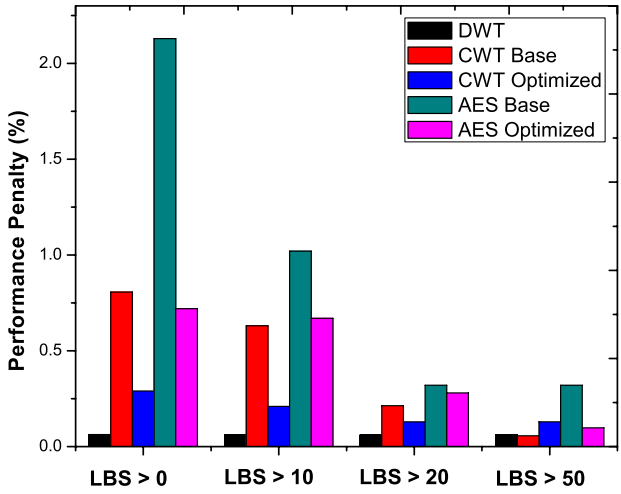


Figure 3.11: MR-IMO exploration of real benchmarks with loop body size (LBS) restrictions on performance penalty trade-offs.

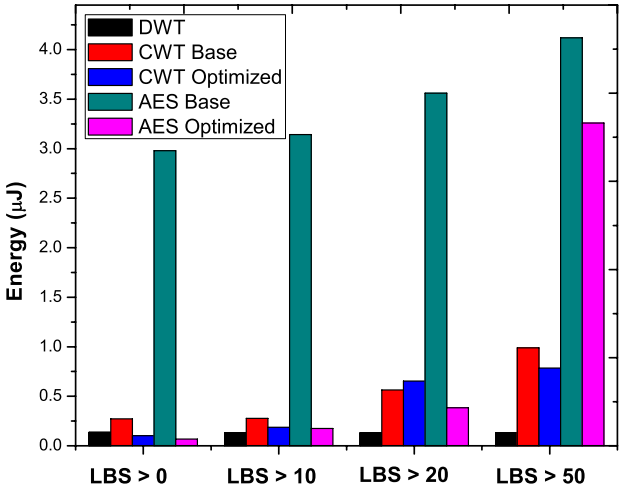


Figure 3.12: MR-IMO exploration of real benchmarks with loop body size (LBS) restrictions on energy consumption trade-offs.

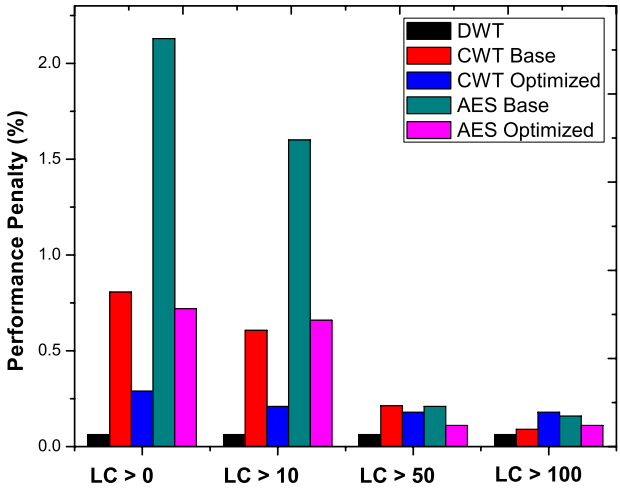


Figure 3.13: MR-IMO exploration of real benchmarks with loop count (LC) restrictions on performance penalty trade-offs.

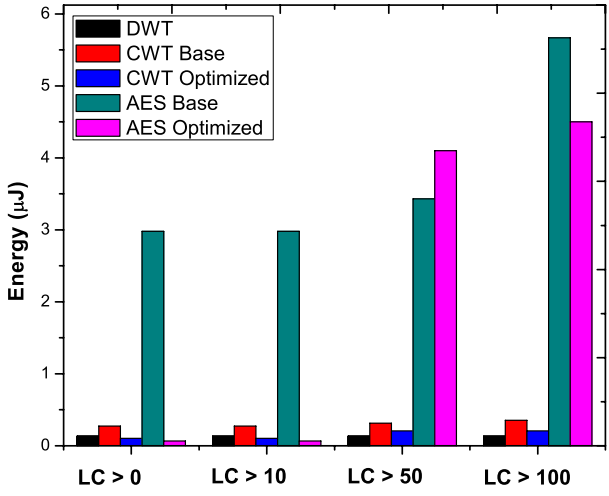


Figure 3.14: MR-IMO exploration of real benchmarks with loop count (LC) restrictions on energy consumption trade-offs.

penalty - energy consumption trade-offs. These particular explorations are subjective to platforms that can tolerate limited hardware costs (wherein MR-IMO becomes a viable option) and applications where there is very little to no tolerance for performance penalty. All the proposed instruction memory organizations directly lead to a clear reduction in both the performance penalty and energy consumption.

In-order to assess the practicality of our assumptions regarding the write cycles in our target applications and to get an idea about the worst case scenarios for the proposed architectures, we compare the total energy consumption of EPOR-IMO and EPOS-IMO. The write energy consumption (WE) for OxRAM based L0 loop buffer per access is assumed to be in multiples of the read energy per access (RE). We takes 5 different cases, where the write energy per access is assumed to be 10, 50, 100, 200 and 500 times the read energy per access. Figure 3.15 details the relative total energy consumption of EPOR-IMO for the different write energy cases across the benchmarks. It's abundantly clear from the graph that our assumptions hold true even when the write energy per access is 500 times more than that of the read energy per access for EPOR-IMO. AES base has the maximum number of dynamic instructions to be read and largest Read:Write ratio. This translates to the negligible increment even when the write energy is 500 times the read energy for OxRAM memories. It's exactly the opposite for AES-optimized. In figure 3.16, we compare the total energy consumption of MR-IMO, EPOR-IMO and LB-IMO for when the write energy is 100 times the read energy (reasonable assumption for 1T-1R OxRAM according to [45]). The effect of lesser writes to the intermediary structures and the even the L0 in case of LB-IMO is clear from this. For applications/benchmarks with a very reduced read:write ratio, the LB-IMO and other simpler organizations become a more viable option as compared to EPOR-IMO. The excess writes to all the intermediary structures will eventually add up in such a scenario and the simpler structure will have a lower total energy consumption.

3.7.2 Exploration by means of synthetic benchmarks

In order to explore the behaviour of our proposed architectures and to probe the effects of the OxRAM based memories more extensively we have carried out a number of simulations on synthetic benchmarks. These synthetic benchmarks are generated by means of a simple C-based tool and are highly customizeable (the length of the instruction code,the number of loops, loop count and loop body size can all be varied according to our requirements). The first experiments carried out are to study the effects of loop body size and loop count on the energy consumption and performance overhead. We chose the

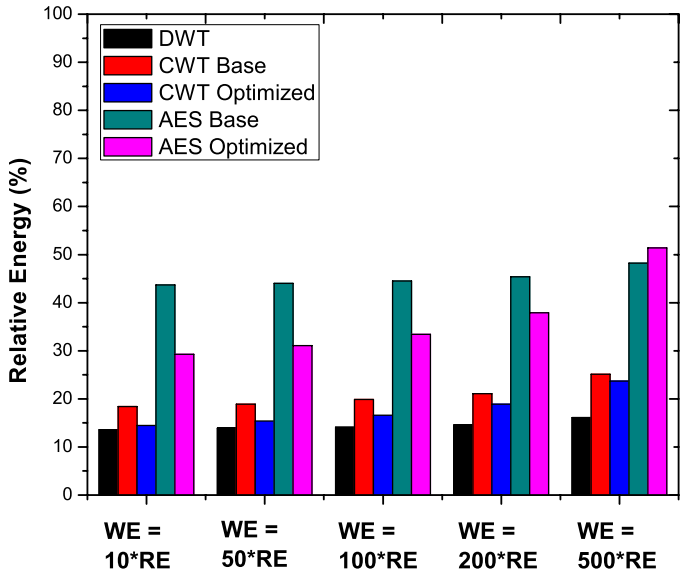


Figure 3.15: Relative total energy consumption of EPOR-IMO for the different write energy cases across the benchmarks. The total energy consumption of EPOS-IMO across all the benchmarks is taken as the base (i.e. 100%)

Modified ReRAM Instruction Memory Organization (MR-IMO) for its obvious susceptibility to both changes in performance and energy with respect to changes in the instruction code.

We can observe the details of the variation in the performance overhead in Fig. 3.17 and for energy consumption in Fig. 3.18 of MR-IMO with changes in loop count (LC) and loop body size (LBS). Here, the synthetic benchmark comprises of a single loop. The proportional increase in loop count greatly contributes to a proportionally reduced performance penalty. The perceived decrease in performance penalty is simply based on the correlation between the loop buffer width and loop body size (loop buffer with = 8 words). In-case of energy consumption however, there is a proportional and predictable increase in the energy consumption with both proportional increases in both loop body size and loop count. In figure 3.19 the same exploration is detailed, but for EPOR-IMO. Although, the magnitude of the energy consumed is lesser than that of MR-IMO (owing to the more energy efficient organization), the energy consumption predictably follows the same pattern as that of MR-IMO.

To further expand our understanding of the contribution of the different

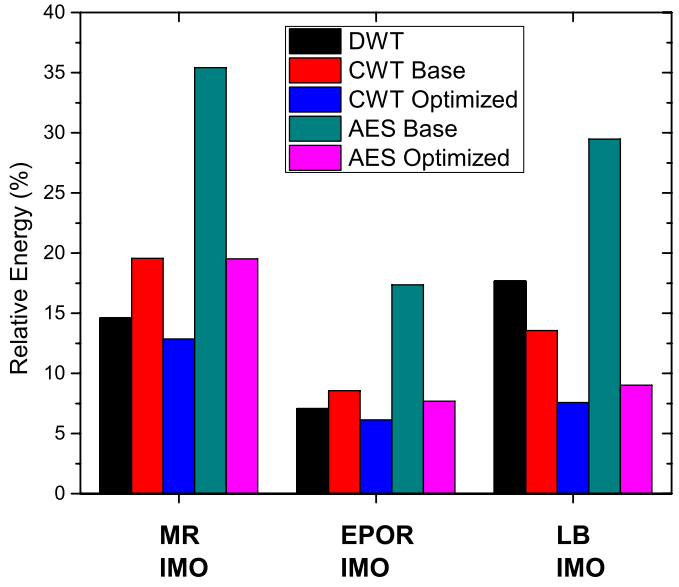


Figure 3.16: Relative energy consumption of the different IMOs for WE = 100*RE. The total energy consumption of BS-IMO across all the benchmarks is taken as the base (i.e. 100%).

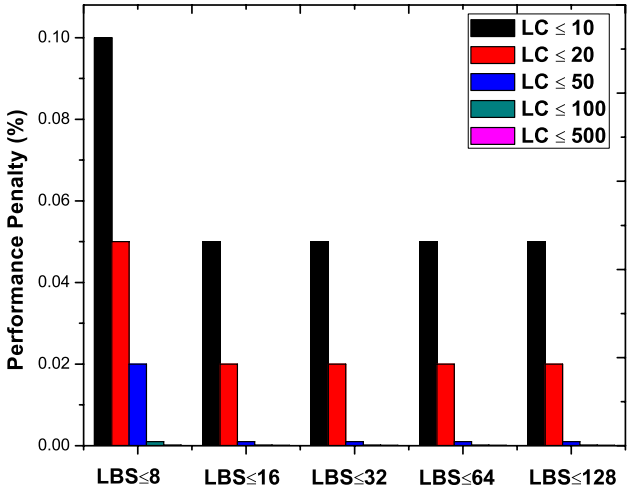


Figure 3.17: MR-IMO exploration with changes in loop count (LC: iteration) and loop body size (LBS) on performance penalty variation.

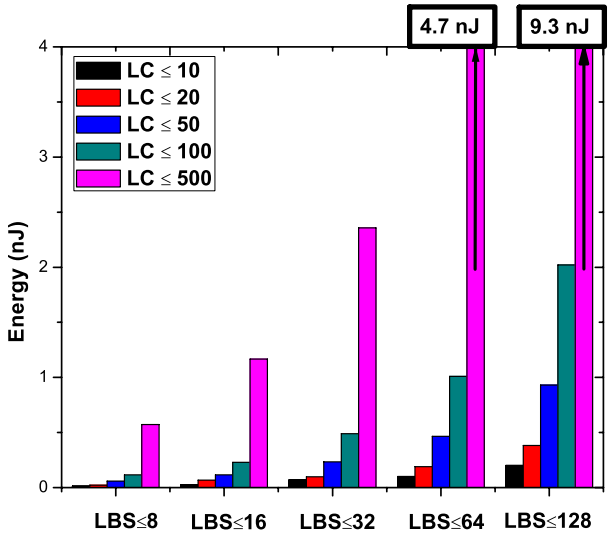


Figure 3.18: MR-IMO exploration with changes in loop count (LC: iteration) and loop body size (LBS) on energy consumption variation.

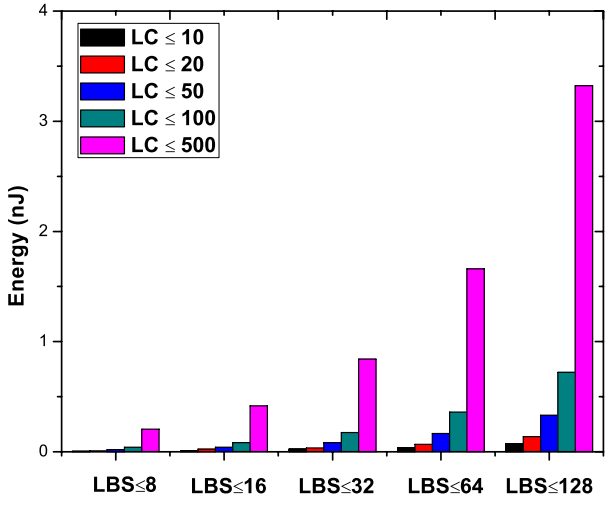


Figure 3.19: Energy consumption variation with changes in loop count (LC: iteration) and loop body size (LBS) for EPOR-IMO. NOTE: No performance penalty is induced here!

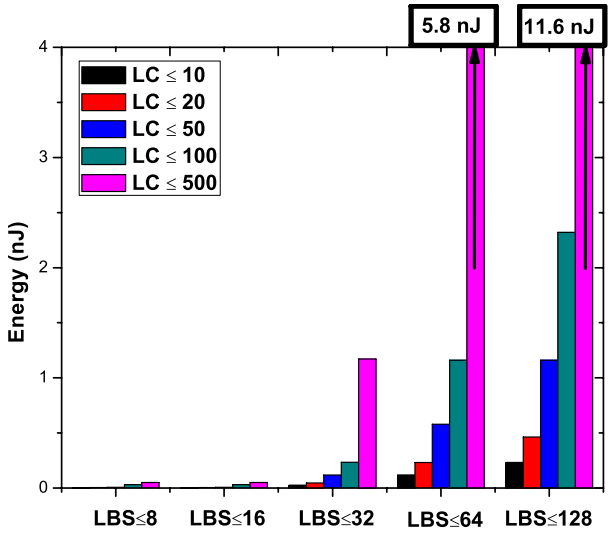


Figure 3.20: Energy consumption variation with changes in loop count (LC: iteration) and loop body size (LBS) for LB-IMO. NOTE: No performance penalty is induced here!

components to the energy consumption, we carried out the same experiment with the Loop Buffer devoid Instruction Memory Organization (LB-IMO), (Fig. 3.20). Again, as expected the energy consumption increases proportionally with proportional increases in LBS and LC. The interesting point to be noted here however, is the variation in energy consumption as compared to MR-IMO and EPOR-IMO. The energy numbers are clearly lesser until the LBS exceeds the width of the VWR. This can attributed to the fact that reading from the VWR is clearly much more efficient for the system as a whole. However, this is not always feasible and when the loop body size exceeds the VWR width and the loop has to be read from the L1, the energy consumption starts increasing exponentially. Hence, the LB-IMO approach is feasible only in case of short loops. Thus we can conclude that LB-IMO is suited for applications that have very small loops with a large number of iterations,and require simplicity in the routing and layout.

In order to study the effects of wider interfaces, we repeat the above experiments for a wider memory organization. Fig. 3.21 and Fig. 3.22 detail the changes in the performance overhead and energy consumption respectively of MR-IMO architecture with wider interfaces (MR-IMO wide), with changes in loop count (LC) and loop body size (LBS). Fig. 3.24 and Fig. 3.23 detail

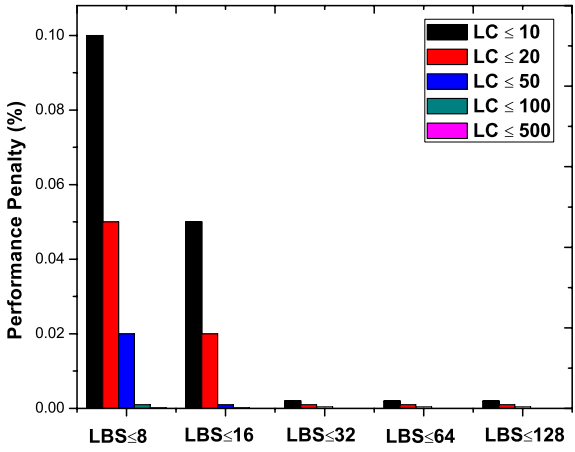


Figure 3.21: MR-IMO (wide) exploration with changes in loop count (LC: iteration) and loop body size (LBS) on performance penalty variation.

the same experiment with wider interfaces for the LB-IMO (LB-IMO wide) and EPOR-IMO (EPOR-IMO wide) architectures. The increase in width here corresponds to a two fold increase in the line size of each element in mentioned configurations (LI, VWR/s, L0 and the L0 line register). The figures clearly show a decrease in performance penalty and energy consumption with wider interfaces and expected trends as before. This can clearly be attributed to more data being read from the VWR and other smaller elements. While, this is certainly advantageous from the energy consumption point of view, it is also limited by a number of factors. The area consideration, layout and routing problems created by the very wide interfaces, and finally the increased access delay due to larger capacitive load on the word lines for L1 and L0 are some of the main areas of concern.

3.8 Conclusions

A number of significant conclusions can be drawn from the above results. Firstly, transition to NVM based memories is almost always advisable for the target applications for ultra low power embedded systems. The wide word access for the L1 and L0 memories and their coupling with the

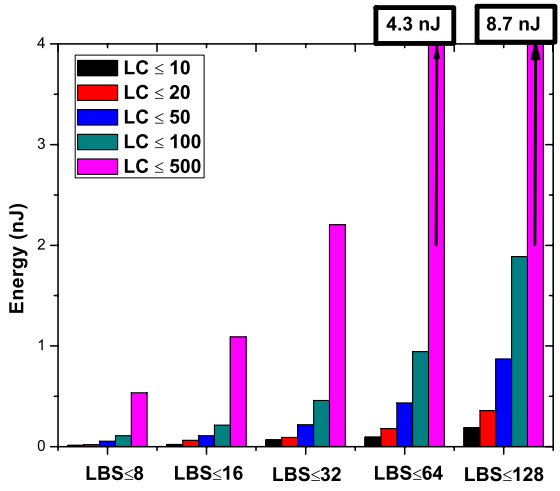


Figure 3.22: MR-IMO (wide) exploration with changes in loop count (LC: iteration) and loop body size (LBS) on energy consumption variation.

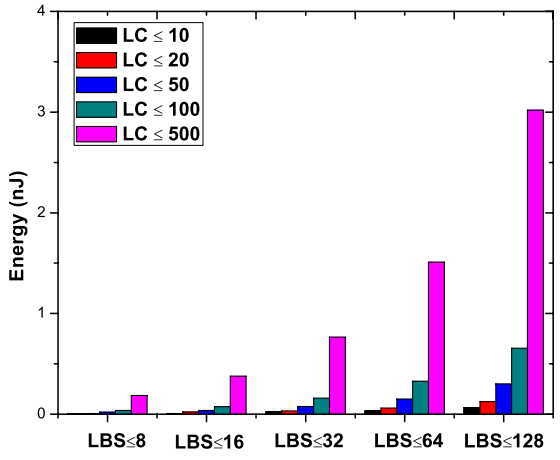


Figure 3.23: Energy consumption variation with changes in loop count (LC: iteration) and loop body size (LBS) for EPOR-IMO (wide). NOTE: No performance penalty is induced here!

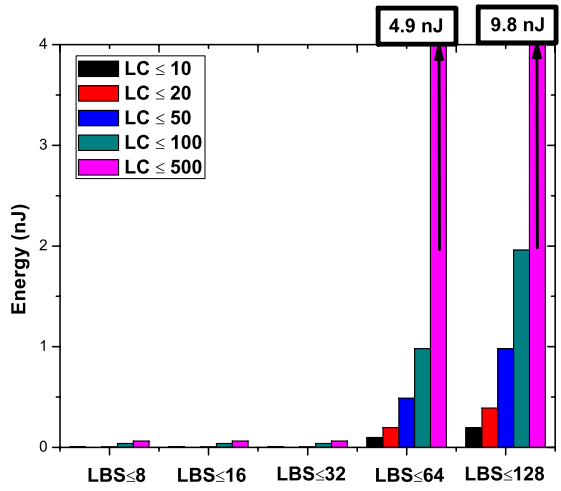


Figure 3.24: Energy consumption variation with changes in loop count (LC: iteration) and loop body size (LBS) for LB-IMO (wide). NOTE: No performance penalty is induced here

VWRs, along with dynamic instruction mapping into these elements ensures lesser write cycles into the OxRAM. This increases the lifetime, while reducing the total energy consumption by a large margin, making the extra hardware (multiplexers and VWR) cost justifiable. The smaller size of the OxRAM cell and array compared to SRAM also makes it easier to tolerate the extra hardware costs. The L0-line register addition ensures that the performance of the system is not compromised in the most optimal instruction memory organization (0% performance overhead). The exploration of the different ReRAM based memory organizations, using both synthetic and real benchmarks, shows us that the loop buffer becomes more significant with increasing loop iterations and loop body size.

All the proposed architectural modifications decrease both the performance penalty and energy consumption. The most energy efficient instruction memory organizations are quite clearly EPOR-IMO (Fig. 3.8) and LB-IMO (Fig. 3.9). However, like it has been concluded earlier by means of exploration through the synthetic benchmarks, the LB-IMO organization is limited by the fact that it becomes inefficient for instruction codes with large loops (see Section 3.7.2). This along with considerable margin of reduced energy consumption makes it worth-while to pursue the implementation of EPOR-IMO for the target

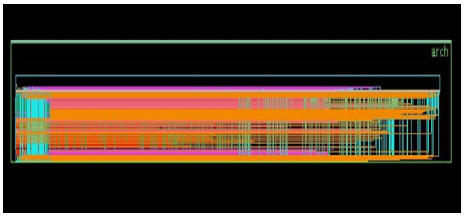


Figure 3.25: Default design floorplan without optimization.

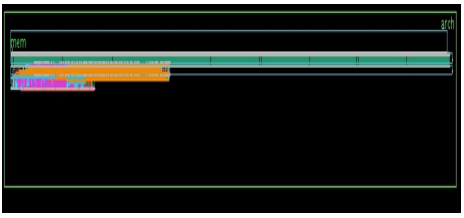


Figure 3.26: Optimized floorplan.

applications and other similar cases in ultra low power embedded platforms despite the extra hardware cost. Like it has been mentioned earlier, these proposals are not limited to a ReRAM based instruction memory organization and can also be suitably applied to STT-MRAM providing it's characteristics meet certain criteria (like reasonable R-ratio, small read latency and read energy etc).

It has to be mentioned that any strain on the floorplan can be avoided by means of smart placement and routing. One way of accomplishing this is by means

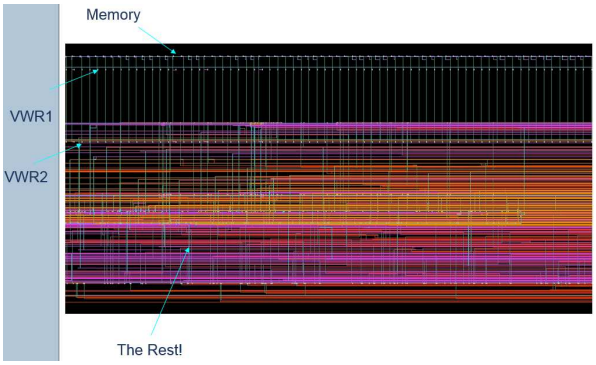


Figure 3.27: VWR and memory routing.

of interconnect friendly floorplans. This includes avoiding long and messy interconnects as much as possible, having distributed instruction memories and controllers and finally the use of irregular architecture (separate interfaces for FG memory - Datapath and FG - BG memory) (reflected in the ISA!). Figures 3.25 and 3.26 detail this.

Chapter 4

eNVM based Configuration Memory Organisation

4.1 Introduction

The previously highlighted ever increasing demand for high performance, computational power, and flexibility from modern systems can be tackled on two forefronts: the processing environment and the memory. The challenges that have to be dealt with on the embedded memory front have been touched upon earlier (Chapter 1) and the proposed directions for finding appropriate solutions, like utilization of NVMs, were highlighted. Keeping these proposals in mind it also makes sense to look at a co-optimization of improved processing environments and memories. Earlier approaches to processing like general purpose processors, digital signal processors or Application Specific Integrated Circuits (ASICs) no longer fit the requirements. They either fail to deliver the required flexibility or exhibit too much area and energy overhead or simply cannot deliver the required performance. Coarse Grained Reconfigurable Architectures (CGRAs) ([194], [126], [77]) have shown their ability, to close a part of the gap for this important application domain together with Domain Specific Instruction Processors (DSIPs). CGRAs exploit the data flow dominance and offer more parallel resources. Unlike the design of a general purpose processor, DSP, or FPGA, the specification and design of a CGRA is heavily influenced by the target application domain.

These architectures usually include a general purpose processor (either RISC based or VLIW) along with a reconfigurable array of cells which speeds up

data flow based computations significantly. These instruction programmable (but more dedicated) cells or functional units (FUs) are interconnected by means of a mesh style network. Register files are distributed throughout the CGRAs to hold temporary values and are accessible only by a subset of FUs. The FUs can execute common word-level operations, including addition, subtraction, and multiplication. Programming the cell matrix requires specific memory organizations that efficiently enforces compiler decisions for every and every cell. This usually implies reading/writing very wide words from memory. Configurable computing is a subset of spatial computing that provides computation and communication resources that can be configured, or programmed, at load time or run time. FPGAs are a common example of a load time configurable computing system, while CGRAs such as [36] and [54]] are examples of run time configurable computing systems. In contrast to FPGAs, CGRAs have short reconfiguration times, low delay characteristics, and low power consumption as they have much lesser wire/interconnect overhead and are constructed from standard cell implementations. Thus, fine-grain gate-level reconfigurability is sacrificed, but the result is a large increase in hardware efficiency.

In this chapter, we extend the wide word OxRAM array presented earlier in Chapter 3 for explorations on the configuration memory organization for CGRA interfaces. These interfaces are optimized for the large configuration words that are particular to CGRA architectures [1]. For our explorations we have utilized the ADRES framework developed at imec [132]. This affects design of the CGRA interface design significantly. We replace the traditional SRAM based interface by a eNVM based alternative. The eNVM based CGRA interface also re-utilizes the architectural modifications proposed in 3.5.1 to make NVM based architectures viable. This chapter does not focus on the endurance or reliability aspects of the eNVM, which is a distinct and separate research matter to be dealt with. The rest of the chapter is organized as follows: Section 4.2 discusses similar work being carried out. Section 4.3 presents the original base architecture for our test-benches. In section 4.4, the different eNVM based configuration memory interfaces that are proposed as alternatives are detailed. The various components involved and the motivation behind their use is explained. We analyze write access patterns of the embedded application benchmarks and discuss the insights that led to the proposed interfaces. The energy modeling and, circuit and technology level considerations are briefly explained in section 4.5. Section 4.6 shows the architectural evaluation results of the proposed methodologies and compares them against the base-line SRAM architecture. Even in the worst case scenario's the most optimal eNVM based CGRA interface showed about 75% reduction in read energy consumption at 0% performance penalty.

4.2 Related Work

The work presented in [154] outlines and evaluates the potential application of emerging non-volatile memory technologies to reconfigurable architectures. The feasibility of architectural integration and innovations in reconfigurable architecture for non-volatile memories are also discussed. A novel reconfigurable hybrid cache architecture (RHC) is presented in [24]. Here, the NVM is incorporated in the last-level cache together with SRAM. RHC can be reconfigured by powering on/off SRAM/NVM arrays in a way-based manner. Hardware-based mechanisms to dynamically reconfigure RHC on-the-fly based on the cache demand are proposed. Experimental results on a wide range of benchmarks show that the proposed RHC achieves an average 63%, 48% and 25% energy saving over non-reconfigurable SRAM-based cache, non-reconfigurable hybrid cache, and reconfigurable SRAM-based cache while introducing at most 4% performance overhead over non-reconfigurable SRAM-based cache and non-reconfigurable hybrid cache.

In [225], bandwidth-aware re-configurable cache hierarchy (BARCH) with hybrid memory technologies is proposed. BARCH consists of a hybrid cache hierarchy, a reconfiguration mechanism, and a statistical prediction engine. The hybrid cache hierarchy chooses different memory technologies to configure each level so that the bandwidth provided by the overall hierarchy is optimized. Compared to traditional SRAM-based cache designs, the proposed design improves the system throughput by 58% and 14% for multi-threaded and multiprogrammed applications, respectively.

In [171], the main memory system consisting of PCM storage is coupled with a small DRAM buffer to avail the latency benefits of DRAM and the capacity benefits of PCM. [169] proposes an efficient hard-error resilient architecture that allocates error correction entries in proportion to the number of hard-faults in the line. Another interesting proposal [168] for an architectural technique exploits the asymmetry between the SET and RESET operation of PCM devices to pro-actively SET all the bits in a given memory line well in advance of the anticipated write to that memory line an thus reduce the effective write latency.

Unlike most of the work presented above our architecture is primarily a software controlled configuration memory leading to a lower energy and area overhead. Most of the related work carried out on utilizing NVMs instead of traditional memory solutions has been focused on the Data Cache, and solutions for the Instruction memory (specifically CGRA based) may require

different architectural decisions. More importantly, the highest levels that are frequently accessed are not SRAM based in our proposals unlike that of the above given references. The works presented above are orthogonal and follow a path rather complementary to our proposals.

4.3 CGRA Basics and Base architecture

CGRAs focus on the efficient execution of the type of loops. By neglecting non-loop code or outer-loop code that is assumed to be executed on other cores, CGRAs can take the VLIW principles for exploiting ILP (Instruction Level Parallelism) in loops a step further to consume less energy and deliver higher performance, without compromising on available compiler support. Higher performance for high-ILP loops is obtained through two main features that separate CGRA architectures from VLIW architectures. First, CGRA architectures generally provide more Issue Slots (ISs: logic on which an instruction can be executed, typically one per cycle like ALUs/FUs etc) than typical VLIWs do. The higher number of ISs directly allows to reach higher IPCs, and hence higher performance. Secondly, CGRA architectures typically provide a number of direct connections between the ISs that allow data to 'flow' from one IS to another without needing to pass data through a Register File (RF). As a result, less register copy operations need to be executed in the ISs which in turn frees ISs for more useful computations.

As mentioned earlier, we utilize the ADRES framework for our explorations. ADRES (architecture for dynamically reconfigurable embedded systems) can be tuned to meet both multimedia applications and (wireless) communication requirements. The ADRES consists of a tightly coupled very long instruction word (VlIW) processor and a coarse-grained reconfigurable array. ADRES is a flexible template to generate concrete instances. Together with a re-targetable simulator and ANSI-C compiler, this tool chain allows architecture exploration and development of application-domain specific processors. It's key features are as follows:

- high performance thanks to multiple levels of parallelism:
  - instruction-level parallelism
  - multi-threading support
  - vector instructions support
- extremely low power by avoiding central components and through optimization of functional units, and data and instruction memory

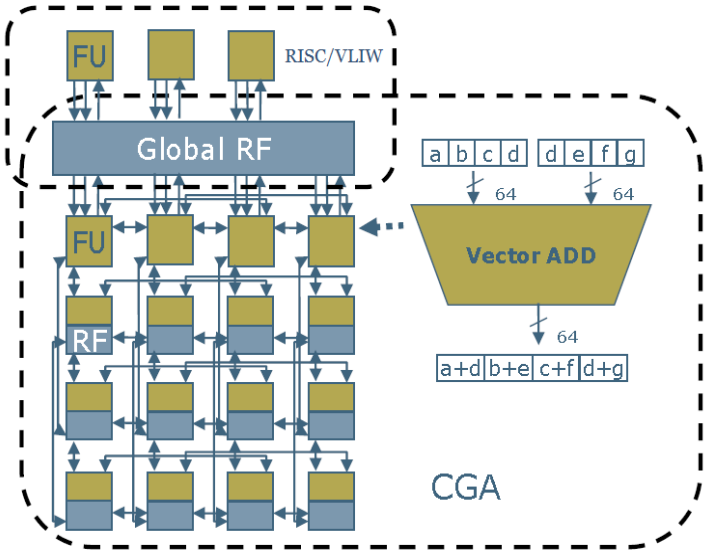


Figure 4.1: Architecture template for a CGRA based system.

- C-programmable

The use of the ADRES simulator framework for our system level explorations was motivated by the target applications (wireless and multimedia), our exploration space (instrcution memory system architecture for reconfigurable systems) and the in-house availability (of a real system).

An illustration of a representative template [1] for a CGRA based system is shown in Fig. 4.1. It consists of an array of basic components, including Functional Units (FUs), Register Files (RFs) and routing resources. At the top level, it tightly couples a VLIW/RISC processor and a CGRA in the same physical entity. The computation intensive kernels of the application, typically loops, are mapped onto the re-configurable array, whereas the remaining code is mapped onto the processor. The data communication between the processor and the reconfigurable array is done through the shared RF and shared memory access.

The execution of the CGA is controlled by the CGA control unit (as mentioned earlier), while the VLIW/RISC processor has a standard Fetch-Decode-Execute pipeline, that is fed by an instruction cache. The configuration memory is loaded (after system reset) through the configuration memory interface.

A central control unit manages all CGA control units. CGA control units provide configurations and control (synchronization) signals to different units of the CGA array within one partition. Similar to the 'Instruction Memory Organization' (IMO) that serves as the processor interface, in many of the commercially available embedded systems today, the interface to the processor/CGRA (Fig. 4.2) consists of two levels: the configuration memory (Level 1; L1) and the configuration cache (Level 0; L0). The core contains as many CGA control units as there are CGA partitions. The configuration memory is an ultra wide memory, and larger than the configuration cache (about 8-16 times), which is closer to the data-path. The cache implementation is meant to reduce the energy consumption of the control unit.

This configuration cache effectively works as a traditional L0 loop buffer [205], which is used to hold frequently executed program loops. As it has been mentioned earlier, this is fairly common in processors meant for loop-dominated applications (more hardware controlled flavours of the loop buffer are the loop cache [56] or the filter cache [99]). Here, in the first instance of a loop, configuration words from the configuration memory are fetched and copied to the configuration cache, and these configuration words are then used in subsequent loop calls from the configuration cache. In our particular domain, the mapping of code to L1 is static such that it holds all the code that the CGA is ever to execute. The L0 must dynamically fetch from L1 whatever must be executed at a given time. When executing from the configuration cache, the configuration memory can remain idle, which has a positive impact on the system power consumption ([10], [12]). This particular implementation comes closest to the software controlled clustered loop buffer presented in [81].

The original CGRA interface taken as the base for our target applications and test-benches is detailed in Fig. 4.3. From here on, this interface will be referred to as the **Base SRAM CGRA interface** (BS-CI). The configuration memory is SRAM based, whereas the configuration cache is an array of n D-latch registers and 2 flip-flop registers, where n = CGA cache depth. In the rest of the paper, the configuration cache will always be assumed to have cache depth of 16 words. Every configuration memory line is a single configuration word wide, and the configuration word size is 135 bits. The cache is filled during the first iteration of the loop and used in all the later implementations. A standard cell based implementation is used instead of memory macros for the cache since standard flip-flops and D-latches consume significantly less energy when being read. We start from the architecture detailed in Fig. 4.2 as baseline, realized with SRAMs. a simplified diagram detailing this is given in Fig. 4.3.

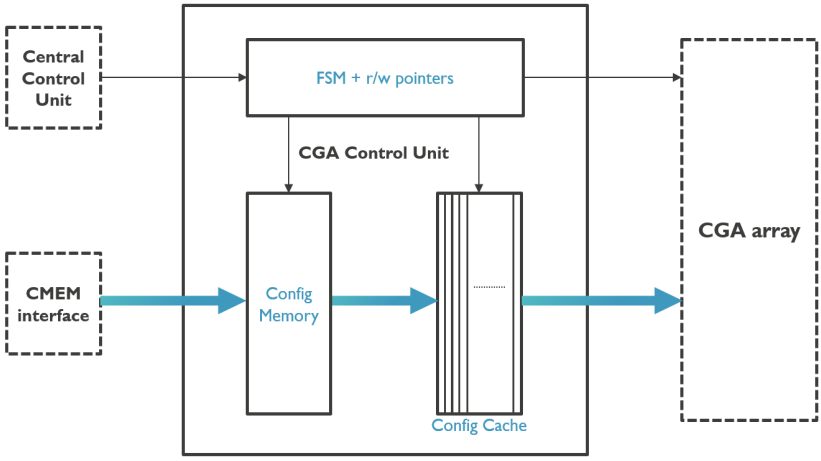


Figure 4.2: The different building blocks of a CGRA interface.

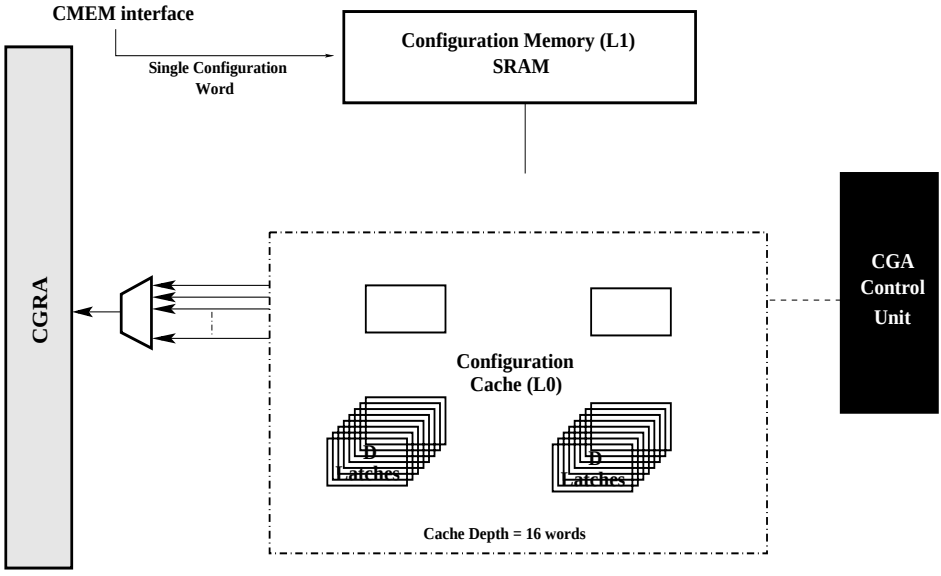


Figure 4.3: SRAM based initial CGRA interface: BS-CI.

4.4 eNVM based memory architecture exploration

The loop dominated nature of the codes for our target applications leads to more read accesses as compared to write accesses. This read-write asymmetry leads to significantly more usage of the configuration cache. This is certainly advantageous considering energy consumption (and also performance point of view since the cache here is much closer to the data-path than the configuration memory). We have considered OxRAM as the eNVM of choice for configuration memory explorations presented here. The terms OxRAM and ReRAM will be used interchangeably in this chapter. The low energy read access of the OxRAM and the application-wise read-write asymmetry makes the use of OxRAM highly preferable for embedded systems running the target applications. Other than the obvious advantages of area and non-volatility. The compatibility of OxRAM with logic technology also makes it suitable to replace SRAM at such high levels.

Thus in this section we explore several CGRA interfaces envisaging the substitution of SRAM based memory systems by a OxRAM based counterpart. Given the high sensitivity of the configuration cache (loop buffer) to the increased write latencies, we will carefully explore this L0 layer in order to attain energy savings with no performance loss. A OxRAM based loop buffer with specific architectural extensions will be considered as a replacement to the baseline configuration cache. First, we propose some internal architecture modifications to make OxRAM modules achieve higher bandwidth at low energy costs. Later, we will present different CGRA interfaces as alternatives to meet the timing requirements while saving as much energy as possible.

4.4.1 OxRAM architectural modifications to address technology limitations

While the read-write asymmetry can alleviate the problems associated with the OxRAM write access to a certain extent, it would still not be a feasible alternative to SRAM based IMOs. The write energy consumption will be manageable as a result of smaller number of write accesses compared to read accesses (read-write asymmetry) with acceptable OxRAM cell write operation and area benefits [59]. However, the penalty due to the OxRAM write latency will still result in performance penalties. We have assumed an OxRAM write latency of 8 cycles compared to the SRAM read latency of 1 cycle; a valid assumption as per [59] and [25]. The OxRAM device and bit-cell are exactly the same as in section 3.3.1.

One of the ways to limit the write access problems is by the use of wide word access schemes. Utilizing wide word access schemes for memories has a number of advantages:

- Writing wide words into the L0 loop buffer reduces the average write energy consumption per bit.
- Wide word write also reduces the effective total time required to write the frequently executed loops into the loop buffer.
- The wide word read access for OxRAM is much more efficient per bit as compared to that of SRAM because the cumulative capacitive load of the SRAM cells is not present.

The above mentioned reasons are sufficient enough to motivate the transition towards a wide word access scheme for the L0 loop buffer.

4.4.2 CGRA interface exploration

Given the high write/read latency ratio (write latency is 8 times higher then read), we can no longer assume that the configuration words are copied from the L1 memory layer to the loop buffer during the first pass of a loop without performance penalty. Thus, we would need to repeatedly read configuration words from the L1 configuration memory while writing into the loop buffer, which will still be a highly energy consuming due to the size of the memory. We utilize the VWR presented in 3.5.2(Fig. 3.5)

A first CGRA interface proposal, named **Modified ReRAM based CGRA Interface** (MRCI), is shown in Fig. 4.4. Both the configuration memory and the loop buffer are OxRAM based wide word access memories: 8 configuration words wide access for L1 memory and 4 configuration words wide access for L0 loop buffer. In the proposed architecture, each VWR cell size has a capacity of 4 configuration words, which is the word-size/output of the VWR and equal to the line size of the L0 loop buffer. The VWR has a single cycle access similar to register files. We write 4 configuration words per write cycle (L0 line size) into loop buffer to minimize the write energy consumption and the performance penalty due to OxRAM write latency.

The configuration line that contains the configuration words to be accessed is always transferred to the VWR in the first pass. In subsequent accesses, data is fetched from the VWR till the loop buffer write cycle is completed. The data is then fetched from the loop buffer till until the next loop is encountered.

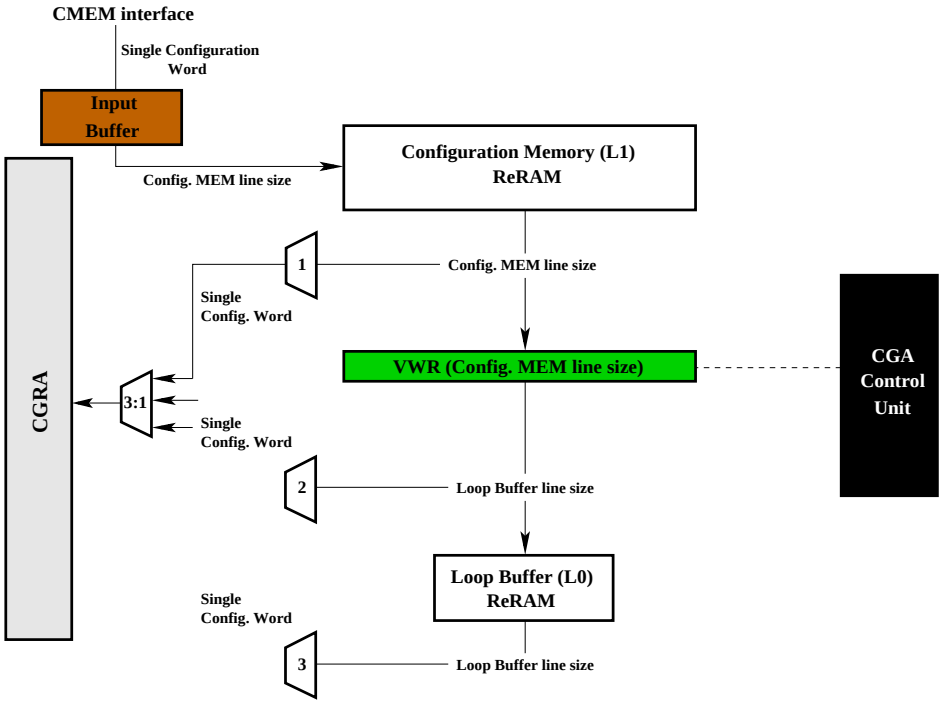


Figure 4.4: Modified ReRAM based CGRA interface: MRCI.

The VWR is completely filled in each of its write cycle. Multiplexer network 1 extracts a single configuration word from wide word written into the VWR towards the processor. Once the *loop flags* are activated, the entire contents of the VWR cell (L0 line size) that contains the 4 loop configuration words in question is copied into the loop buffer.

The write cycle into the OxRAM loop buffer takes place over 8 cycles as mentioned before. Due to the presence of a smart multiplexer network that selects a single configuration word from the 4 configuration words being written into the loop buffer, the data can be read from the VWR while it is being written into the loop-buffer. Multiplexer network 2 extracts a single configuration word from wide word written into the loop buffer from the VWR towards the processor. Multiplexer network 3 extracts a single configuration word from wide word read from the loop buffer towards the processor.

In order to explore the effects of memory width on energy consumption, we implement an interface with reduced word access size, named **Reduced Instruction - Modified ReRAM based CGRA interface** (RI-MRCI).

Here the configuration memory line is 4 configuration words wide and the loop buffer is 2 configuration words wide. Note that the memories in this configuration are still considerably wide (since each configuration word is 135 bits wide) and so this narrower interface makes every single access less energy consuming even though the energy per bit increases. If the extra-wide interface is efficiently exploited (i.e. if we are able to reduce the number of accesses to the large configuration memory by means of the VWR), it is still advisable to opt for wider interfaces as much as possible.

4.4.3 Alternatives to further reduce the energy consumption

Addressing leads to a single cycle delay while the VWR is being updated and the data is to be read from the configuration memory. Due to the large configuration word size, only a few configuration words can be held in the VWR and it has to updated frequently if the number of loop iterations are small. Hence, whenever the loop size spans over the length of the configuration memory line size (or VWR size), it is easier to simply read the second line from the configuration memory itself. This approach would however be less energy efficient due to a significant number of reads from the larger configuration memory. A bypass from the configuration memory to the loop buffer would in-effect reduce the need for updating the VWR frequently and reduce the number of read accesses to the configuration memory. This third alternative to the CGRA interface, named **Bypass - Modified ReRAM based CGRA interface** (B-MRCI), has been highlighted in Fig. 4.5.

Another alternative to the aforementioned problem would be to introduce a delay cycle internally (without stalling the processor. This fourth configuration is termed **Multiple Update - Modified ReRAM based CGRA interface** (MU-MRCI). Now the VWR can be updated every time the loop size extends over the configuration memory line size.

Taking into consideration, the initial D-latch and Flip-Flop based implementation of the configuration cache, it is hard to imagine a OxRAM based loop buffer substitute achieve lower energy consumption (even at extremely lower read access energies..). Hence, a last alternative to the CGRA interface is proposed:**Multiple VWR - Modified ReRAM based CGRA interface** (MV-MRCI). Here, the loop buffer is abandoned and another VWR is introduced (Fig. 4.6). Now, assuming that the number of configuration words in any loop body does not extend over the length of 2 VWR's (16 configuration words), the energy consumption in this case is expected to be the least. This can be attributed to the extremely low energy read access of the VWR and the favourable read access energy of the wide word OxRAM as compared to

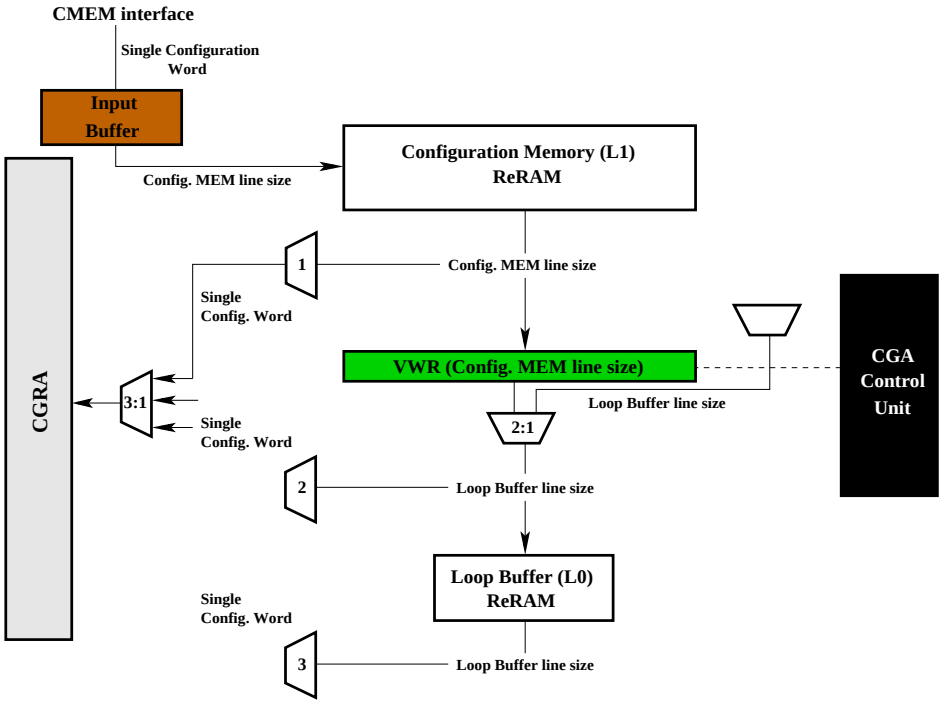


Figure 4.5: CGRA interface with bypass to loop buffer : B-MRCI

SRAM. There is an obvious trade-off here with the length of the loop bodies, but in the application domain and for the target platform considered (CGRAs) under consideration, 16 words is a large enough assumption for most loop nests. Moreover, code transformations are always at hand to limit the body size when required.

In all the different architectural explorations presented above, we have ensured a 0% performance penalty by means of selective mapping into the VWR and Loop Buffer. Transferring entire cache lines also helps mitigate the write latency issues.

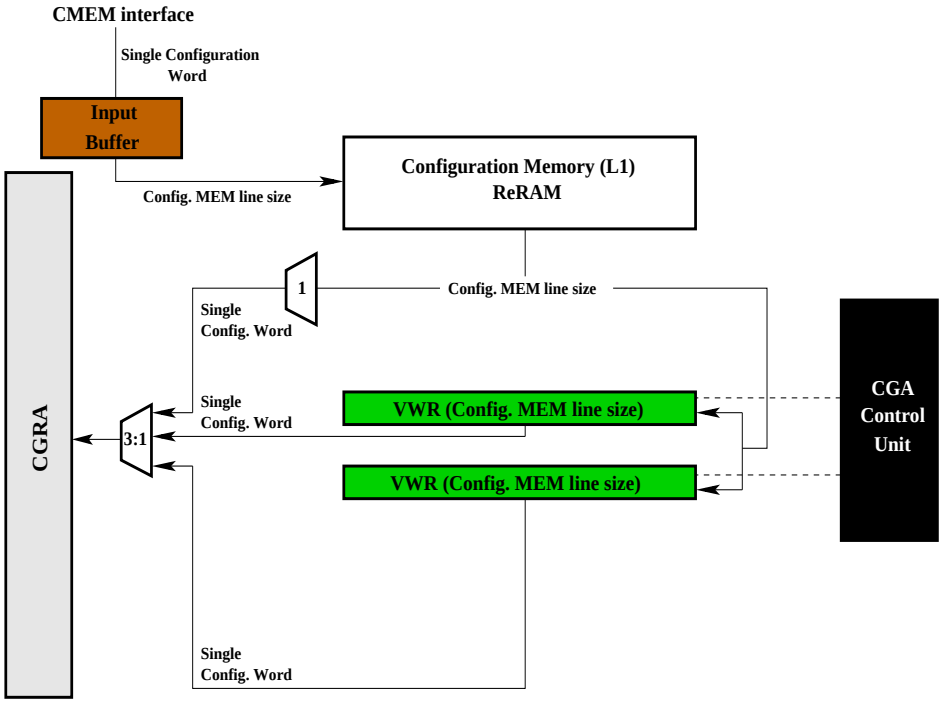


Figure 4.6: Multiple VWR - Modified ReRAM based CGRA interface: MV-MRCI.

4.5 Energy Modeling

4.5.1 SRAM, Configuration cache and VWR energy

The energy numbers are calculated internally based on the PTM cell library for 32nm technology node. V_{dd} is taken as 0.9V for memories and 0.7V for logic. The SRAM based configuration memory has a single read and single write port. The above mentioned technology considerations are extended to all energy calculations for fair comparisons. The VWR is D-latch based in our implementations since the timing is entirely regulated by means of a control unit that regulates a state machine. In this implementation, we never write and read from the VWR simultaneously and hence the D-latch based implementation will suffice instead of the flip-flop based implementation.

4.5.2 OxRAM memory configuration

The OxRAM energies and access latencies are calculated based on a wide word OxRAM array presented 3.3.2 (Fig. 3.2). The wide word access OxRAM array structure is detailed in Fig. 3.2. The configuration memory are is into sleepy modes when it is not being used. The leakage energy values for the SRAM memory, configuration cache and VWR is again obtained by means of internal assessment based on PTM cell library for 32nm technology node. Again, like it has been mentioned in the previous chapter, the numbers presented here were computed at the beginning of the PhD when the OxRAM technology was less mature. This does not match the state of the art data presented in sections 2.3 and 2.4.2. The numbers were based on reasonable OxRAM improvements, voltage scaling assumptions and a planar thick oxide access transistor for the bit-cell derived from the promising data and literature on HfO_2 based OxRAM3.3.1. However, over time it's become clear that OxRAM will require relatively high voltages to switch that demand I/O transistors or a specific selector.

4.6 Results and Discussion

4.6.1 Assumptions

The following assumptions were made in the simulations and tests and they are all explained below in the context of our target domain and paper focus:

- The target applications for such low power embedded CGRA platform are such that the configuration memory is able to accommodate any kernels to be run before-hand and will not be subject to dynamic writing during the execution process. This is similar to the ultra-low power DSP platforms like TI TMS320. Thus, the kernels can be loaded before run-time and the delay due to the slow writing into the OxRAM based configuration memory is avoided.
- Due to the loop dominated codes of our target applications and the wide word access LI-L0, the memory is almost always in a read only memory mode. Thus, while we require every access cycle for the read mode, the duty cycle of the write mode is greatly reduced. So every loop statement is read at least a factor L times compared to the write access count. The writes are slower than the read with a typical factor K, where the OxRAM memory read latencies can easily be assumed to be the same as that SRAM memory. In practice $L >> K$, where K ∼4-8 cycles (based

on [114], [190] and proposed read access models) for every K, so slow write times are acceptable. From the above and the proposed OxRAM memory structure, we can conclude that modeling the read access is sufficient to provide a solid idea about the feasibility of the proposed architectural changes.

- We assume that the data transfer from the configuration memory to VWR and from all components to processor when connected directly or via a multiplexer are single cycle accesses. This is a fair assumption since the VWR has a single cycle access and along with the multiplexers have small logic depth [173].
- We assume that the size of the loop body of all loops present in the configuration code is lesser than or equal to the loop buffer capacity. All the loop configuration words, except for the function calls that lie outside the loop body in question, are transferred into the loop buffer. This is achieved by applying proper code transformations upfront.

A broad range of representative benchmarks have been used to test and compare the energy consumption and performance of the original and modified configuration memory architectures. They are all extracted from a wireless LAN application code base which is compatible with the WLAN standards so they are all fully realistic drivers for our target domain. These are listed below:

- **256 Point FFT**: Fast Fourier Transform for 256 input samples.
- **2k FFT**: Fast Fourier Transform for 2048 input samples.
- **PrecMtx2x2**: MIMO Precoding.
- **cpstIQCFO**: Compensate IQ inbalance and CFO.
- **shflCarrier**: Shift subcarriers.
- **shflCarrierPilot**: Shuffle data subcarriers and pilots.
- **compAngle**: Compute angle for channel interpolation.
- **interpFreq**: Frequency interpolation.
- **invMtxQR2x2**: 2x2 matrix inversion.
- **compDelta**: Compute step size for channel interpolation.
- **compLLRCoef**: Compute LLR scaling coefficients for channel interpolation.

Table 4.1: Dynamic read energy contributions of the different instruction memory organization components.

Benchmark	No. of loops	Loop body sizes	Dynamic configurations	Avg. loop iterations
256FFT	4	6,6,7,10	56	14
2kFFT	6	6,6,6,6,8,5	768	128
PrecMtx2x2	1	8	150	150
cpstIQCFO	1	8	128	128
shflCarrier	1	8	50	50
shflCarrierPilot	2	14,15	100	50
compAngle	1	8	100	100
interpFreq	2	5,10	250	125
invMtxQR2x	2	8,11	300	150
compDelta	1	11	300	300
compLLRCoef	1	12	150	150
procPayload	2	6,11	300	150
cpstCfo	1	5	8	8
tracking	2	5,6	12	6
SoftDemapQAM64	1	6	16	16

- **procPayload**: Process payload.
- **cpstCfo**: Compensate CFO.
- **tracking**: Track channel movement.
- **SoftDemapQAM64**: Soft demodulation for QAM64.

A profile of the configuration codes of the different benchmarks used are given in Table 4.1.

4.6.2 Discussion of results

The comparisons of the read energy consumptions for all the proposed CGRA interfaces over the different benchmarks are outlined form Fig. 4.7 to Fig. 4.21.

All interfaces are modeled using VHDL and RTL level compilations and simulations of complete benchmarks are analysed real-time via ModelSim. The timing, propagation delays and critical path is ensured by means of regression tests and appropriate verification models. The ADRES template system does not tolerate performance overhead in the form of extra cycles and hence it is

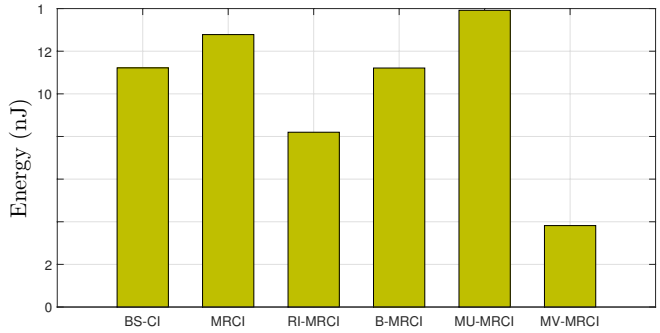


Figure 4.7: 2kFFT benchmark read energy numbers for the different architectural variations.

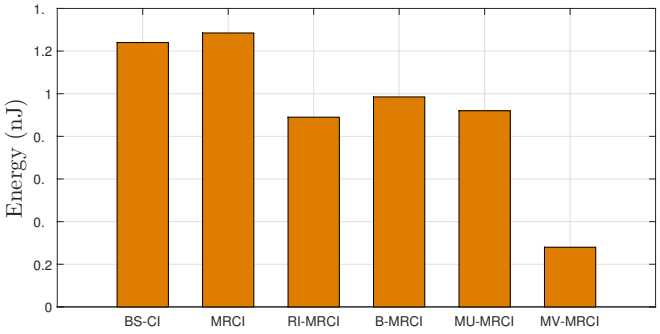


Figure 4.8: 256FFT benchmark read energy numbers for the different architectural variations.

necessary to keep the performance penalty to 0%. The access count and timing analysis confirm that the proposed alternative CGRA interfaces (MRCI, RI-MRCI, B-MRCI, MU-MRCI and MV-MRCI) take the same number of cycles to execute the benchmarks as that of the original CGRA interface (BS-CI). Hence, the proposed CGRA interfaces are able to successfully hide the OxRAM write latency and show 0% performance penalty.

The read energy consumption numbers throw up some interesting results though. The original CGRA interface (BS-CI) is already highly optimised due to the latch and flip-flop based configuration cache. From the figures it

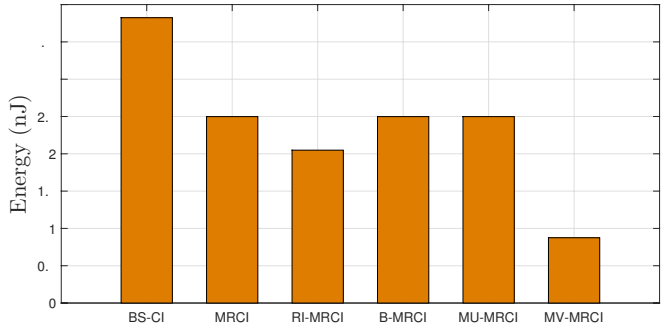


Figure 4.9: PrecMtx2x2 benchmark read energy numbers for the different architectural variations.

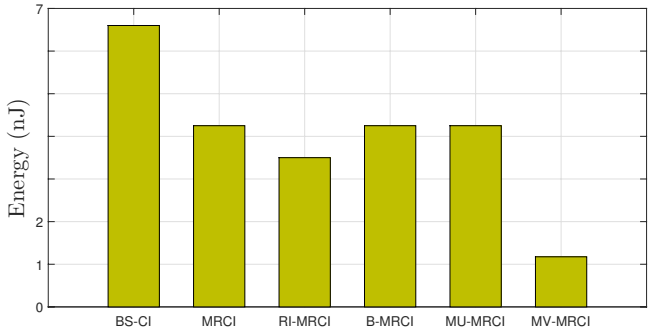


Figure 4.10: cpstIQCFO benchmark read energy numbers for the different architectural variations.

is quite clear that for the benchmarks having larger loops and larger number of loops (like 256FFT, 2kFFT, shflCarrierPilot, invMtxQR2x and interpFreq), the gains from shifting to the proposed CGRA interfaces containing with a LO loop buffer (MRCMO, RI-MRCMO, B-MRCMO and MU-MRCMO) are marginal.

This can be attributed to the highly optimised reference configuration as mentioned before. Reading from a series of latches is bound to be more efficient as compared to reading from a memory array almost always, no matter how energy efficient the technology is. Also, the addressing delay in the template

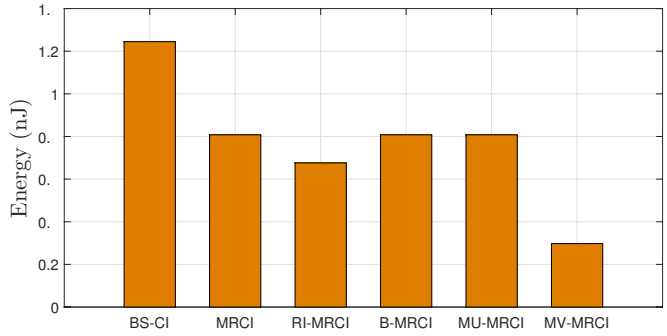


Figure 4.11: shflCarrier benchmark read energy numbers for the different architectural variations.

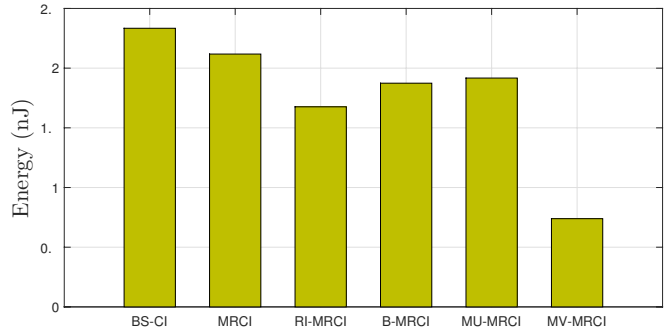


Figure 4.12: shflCarrierPilot benchmark read energy numbers for the different architectural variations.

architecture and the wide configuration words lead to a number of reads from the configuration memory is such scenario's. This affects the energy consumption quite negatively. In-fact, in the worst case scenario's (256FFT and 2kFFT), MRCMO leads to higher energy consumption as compared to the original configuration. This coupled with the fact that RI-MRCMO is more energy efficient for the benchmarks taken into consideration would seem to further the idea that the wide word access OxRAM structure is not optimum for such scenarios. However, like it been mentioned earlier (Section 4.4.1) although the wider interface (comparatively) is theoretically better, but depends on the actual usage and its efficient exploitation. In our set of benchmarks, it is better to go a bit narrower (but wider than usual still).

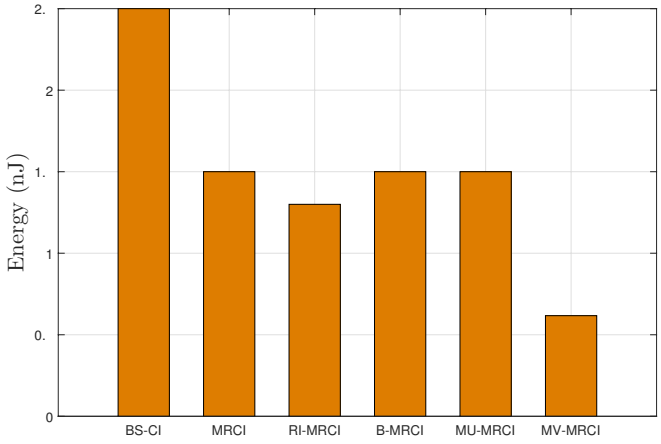


Figure 4.13: compAngle benchmark read energy numbers for the different architectural variations.

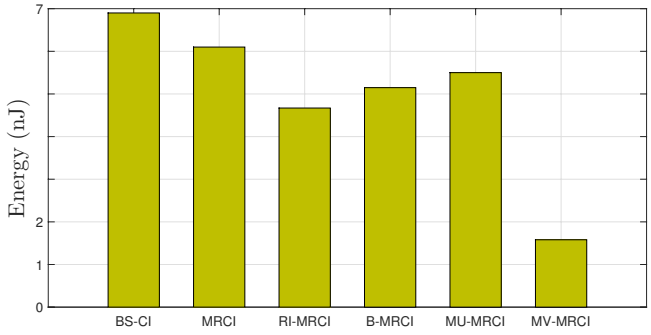


Figure 4.14: interpFreq benchmark read energy numbers for the different architectural variations.

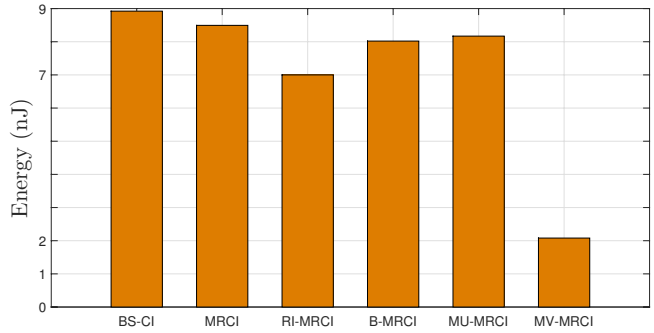


Figure 4.15: invMtxQR2x benchmark read energy numbers for the different architectural variations.

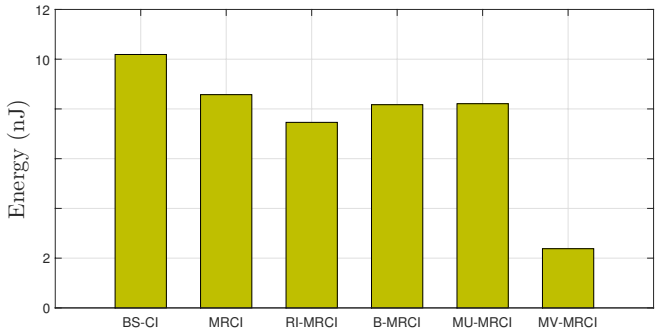


Figure 4.16: compDelta benchmark read energy numbers for the different architectural variations.

A look at the MV-MRCMO energy numbers confirms the fact that wide word accesses are clearly advantageous. Since the highest level here is the VWR, it is a more fair comparison with regards to the said benchmarks. Clearly, the wider access leads to more energy efficiency here. The best case scenario's are the benchmarks that have smaller loops and lesser number of loops with high iterations (leading to more than 75% reduction in energy consumption). The best solution is very specific to the explored domain, where all the loops fit into the two VWR's (for example: compAngle, cpstCfo, SoftDemapQAM64 etc). This structure is more energy efficient and preferable compared to the

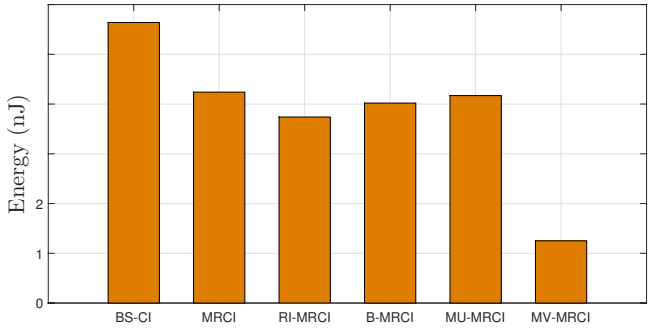


Figure 4.17: compLLRCoef benchmark read energy numbers for the different architectural variations.

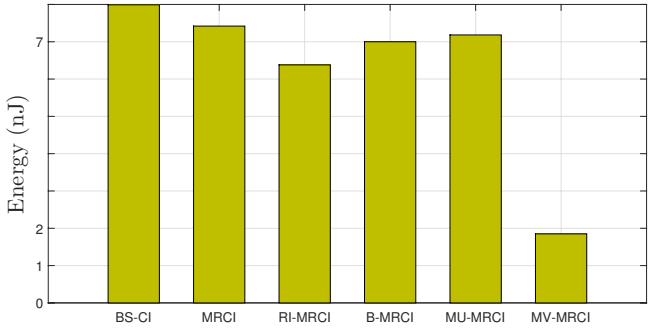


Figure 4.18: procPayload benchmark read energy numbers for the different architectural variations.

FF-latch based original configuration, due to the serial read, structure and implementation of the VWR.

While a OxRAM based conventional loop buffer doesn't offer large gains in energy consumption when compared to the original reference design, it must be pointed out that it is still worthwhile to pursue or explore in other contexts. There are a number of interesting trade-offs that come into play here: the loop body size for example. If the loop body size were to extend over 16 configuration words, the effectiveness of the loop buffer becomes more evident. The base configuration cache or VWR's would be unable to hold the entire loop,

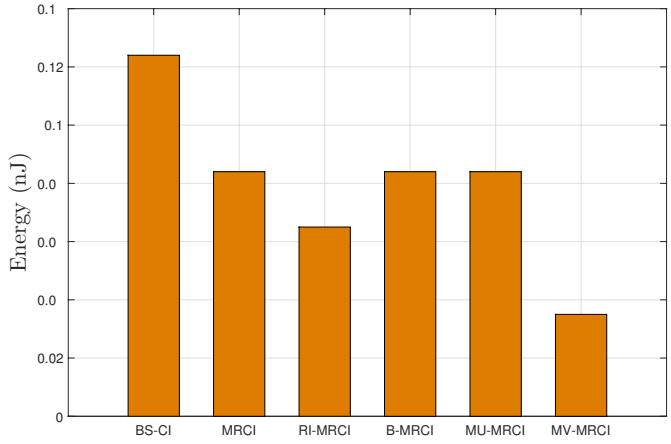


Figure 4.19: cpstCfo benchmark read energy numbers for the different architectural variations.

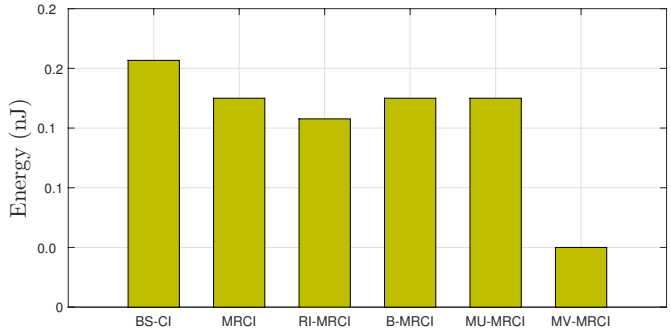


Figure 4.20: tracking benchmark read energy numbers for the different architectural variations.

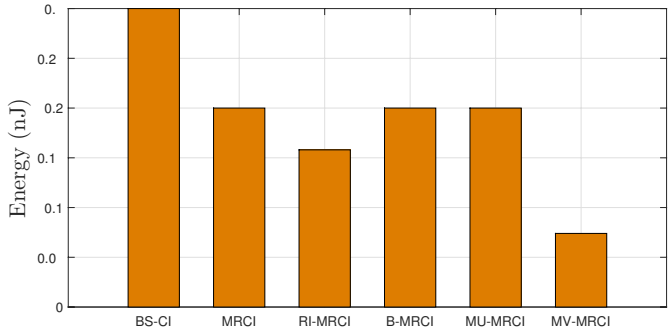


Figure 4.21: SoftDemapQAM64 benchmark read energy numbers for the different architectural variations.

leading to read accesses from the larger configuration memory and thus more energy consumption. The cost-effectiveness of a latch and flip-flop based L0 cache implementation and the wide configuration word context serve to negate the positive aspects of such a wide-word access loop buffer. It could arguably be highly suitable and efficient in simple scalar processors, with regular size instructions, where the full advantages of the wide word access OxRAM array could be exploited.

4.7 Conclusions and Future work

From the results it can be inferred that the proposed substitution of SRAM based L1 configuration memory with a eNVM based alternative has a clear advantage when it comes to energy consumption for the target applications in case of CGRAs. The wide word access for the OxRAM and its coupling to the VWRs, along with dynamic instruction mapping into these elements ensures lesser write cycles into the OxRAM and also in deals with the OxRAM write latency to an extent. It also leads to significantly reduced energy consumption when combined with architectural modifications. The most energy efficient OxRAM based CGRA interface (MV-MRCI) lead a reduction of about 65% read energy consumption even in the worst case scenario's at 0% performance penalty. While the use of the OxRAM loop buffer is not necessarily advantageous in the current context and template, it is worth while to explore it for other commercial scalar embedded instruction memory architectures.

Chapter 5

NVM based I-Cache for High Performance General Purpose Processors

5.1 Introduction

A CPU cache is a hardware cache used by the central processing unit (CPU) of a computer to reduce the average cost (time or energy) to access data from the main memory. The cache is a smaller, faster memory which stores copies of the data from frequently used main memory locations. Most high performance general purpose CPUs have different independent caches, including instruction and data caches, where the cache is usually organized as a hierarchy of more cache levels (L1, L2, etc). All such modern (fast) CPUs (with few specialized exceptions[1]) have multiple levels of CPU caches. The first general purpose CPUs that used a cache had only one level of cache, not split into L1d (for data) and L1i (for instructions). Almost all current CPUs with caches have a split L1 cache and this is generally known as the "Harvard Architecture" after the relay based Harvard Mark-1 computer which introduced it. Split caches help to reduce pipeline bottlenecks as earlier pipeline stages tend to reference the instruction cache and later stages the data cache. Apart from reducing contention for a shared resource, providing separate caches for instructions also allows for alternate implementations which may take advantage of the nature of instruction streaming; they are read-only so do not need expensive on-chip features such as multi-porting, nor need to handle handle sub-block

reads because the instruction stream generally uses more regular sized accesses.

This is followed by the L2 cache. For larger processors, there are L3 and sometimes L4 caches as well. The L2 cache is usually not split and is sometimes shared between cores. The L3 cache, and lower-level caches, are shared and not split. An L4 cache is currently uncommon, and is then on a separate die/even off-chip and generally on non-SRAM. That was also the case historically with L1, while bigger chips have allowed integration of it and generally all cache levels, with the possible exception of the last level. Each extra level of cache tends to be bigger and be optimized differently.

In this chapter, we rethink the highest levels of the SRAM based instruction cache hierarchy and replace it by an NVM based one. We propose a novel instruction cache configuration that address some of the challenges associated with the write behavior of NVMs by means of enhancements to the I-cache Miss Status Handling Register (MSHR). Since the focus is on system level changes, we do not delve deeply into technology and circuit related matters in this chapter. Most of the circuit and system level calculations are based on the use of established simulators. We will focus on single core low power ARM like general purpose platforms here.

The chapter is organized as follows: Section 5.3 motivates the need for architectural modifications in order to replace the SRAM based instruction cache by a NVM counterpart. Section 5.2 discusses similar work and a recent developments with respect to the use of NVMs for caches. Section 5.4 details the proposed architectural modifications to deal with the performance penalty raised by the NVM long write latencies. Section 5.5 presents the results of the proposed methodologies, and a exploration of the effects of the different tunable parameters. Section 5.6 concludes the paper.

5.2 Related Work

As mentioned above, we focus on the highest level instruction cache organization that may or may not require a hybrid structure (with an SRAM). As discussed below, none of the existing published work is directly focusing on that aspect which is our main contribution.

Most of the solutions to the problems faced by these hybrid NVM/SRAM proposals include a combination of software (memory mapping, data allocation) and hardware techniques (registers, buffers, circuit level changes). In [78], the authors propose a data-aware hybrid STT-RAM/SRAM cache architecture which stores data in the two partitions based on their bit counts. To exploit

the new resultant data distribution in the SRAM partition, an asymmetric low-power 5T-SRAM structure is used which has high reliability for majority 'one' data. The proposed design significantly reduces the number of writes and hence dynamic energy in both STT-RAM and SRAM partitions. A write cache policy and a small swap memory to control data migration between cache partitions are also proposed. the evaluation on UltraSPARC-III processor shows that utilizing STT-RAM/6T-SRAM and STT-RAM/5T-SRAM architectures for the L2 cache results in 42% and 53% energy efficiency, 9.3% and 9.1% performance improvement and 16.9% and 20.3% area efficiency respectively, with respect to SRAM-based cache running SPEC CPU 2006 benchmarks.

[116] presents an observation that the intensity of write operations on different cache sets is usually non-uniform for real applications, such as multimedia, multi-programmed, multithreaded applications. The previously proposed hybrid cache schemes can not efficiently and symmetrically utilize the small SRAM to accommodate such widely-existing non-uniform writes on cache sets. Based on this observation, the authors propose a novel hybrid cache design, Dual Associative Hybrid Cache (denoted as DAHYC), as well as the corresponding cache management policy. By organizing the SRAM blocks in the hybrid cache as a semi-independent set-associative cache, several hybrid cache sets can efficiently share and cooperatively utilize their SRAM blocks, instead of exclusively utilizing the SRAM blocks in each cache set in previous hybrid cache schemes, to boost power-efficiency. Through prudently manipulating the locality information of SRAM blocks in both the NVM sets and the SRAM sets, the proposed cache management policy also delivers high-performance. Experimental results show that, compared with previous works, the DAHYC can reduce the dynamic power of the hybrid cache by 24.8% on average and up to 54% for SPEC CPU2006 benchmarks, while at the same time improving the performance of the hybrid cache by 1.16% on average.

The main focus for us lies in energy-sensitive domains like modern micro/warehouse servers or gateway/fog computing nodes. In this particular domain, the ARM platforms are very representative processors. These are based on low-energy low-cost general-purpose processing platforms running embedded applications with real-time constraints. Another aspect to stress again is that while our focus is on the instruction cache, most proposals target data caches. In addition, the highest cache levels that are frequently accessed are still SRAM based most of the proposals currently, whereas the most frequently accessed higher level (L1) instruction cache is explicitly NVM based in our proposal.

5.3 Replacing SRAM with NVMs

We know that SRAM is plagued by leakage problems that worsen as we move towards lower technology nodes (Figure 5.1). This has been highlighted in Chapter 1 and is one of the main motivations for a switch towards NVMs. [187] highlights the leakage trends for High Performance (HP) and Low Operating Power (LOP) CMOS transistors for different technology nodes. The LOP transistors are obviously not as performance oriented as HP. Most of the proposed NVM solutions are low leakage solutions. However, not all NVMs can be considered for L1 caches due to the high performance requirements even though they could be denser and more cost effective per GB. In-fact, Intel and AMD both typically field cache hit rates of 95% or higher. This would lead to a lot of stalls and cause havoc for real-time applications. Thus, NVM technologies such as PCRAM are not considered here, due to its prohibitively high write latency. Low CMOS compatibility and integration problems are some other issues that limit some of the NVMs. These have been highlighted earlier in 2.1. Hence, we look at the two most suitable candidates: ReRAM and STT-MRAM.

Table 5.1 compares the energy consumption of 32KB SRAM, ReRAM and MRAM (STTRAM) cache memories as computed per NVSim [46] for the 32nm tech node with HP transistors. While this is already an old technology, it is the most recent one for which we can obtain realistic values for OxRAM and MRAM from NVSim and other academic sources. NVSim, is a circuit-level beahvioural model suited for an abstraction of NVM performance, energy, and area estimations. It supports various NVM technologies, including STT-RAM, PCRAM, ReRAM, and legacy NAND Flash. The NVSim tool can estimate the access time, access energy, and silicon area of NVM chips with a given organization and specific design options before the effort of actual fabrications. NVSim by using the same empirical modeling methodology as CACTI but starting from a new framework and adding specific features for NVM technologies.

Compared to CACTI, the framework of NVSim includes some additional features like:

- It allows us to move sense amplifiers from inner memory subarrays to the outer bank level and factor them out to achieve overall area efficiency of the memory module.
- It provides more flexible array organizations and data activation modes by considering any combinations of memory data allocation and address distribution.

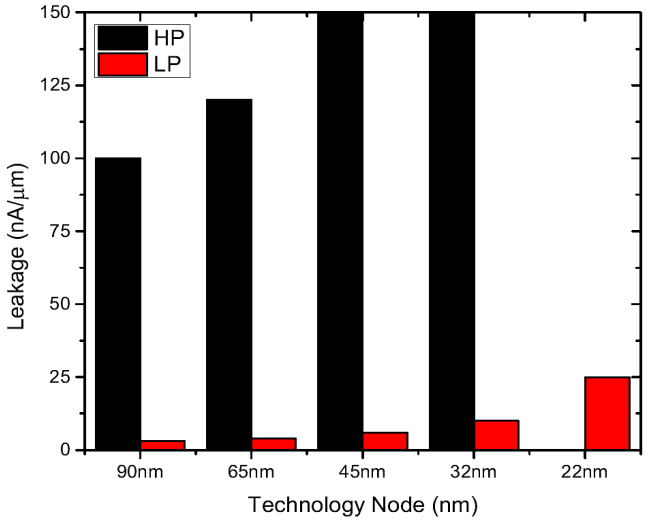


Figure 5.1: SRAM leakage trends for High Performance (HP) and Low Power (LP) transistors for different technology nodes [187].

- It models various types of data sensing schemes instead of voltage-sensing scheme only.
- It allows memory banks to be formed in a bus-like manner rather than the H-tree manner only.
- It provides multiple design options of buffers instead of latency-optimized option that uses logical effort.
- It models the cross-point memory cells rather than MOSaccessed memory cells only.
- It considers the subarray size limit by analyzing the current sneak path.
- It allows advanced target users to redefine memory cell properties by providing a customization interface.

The use of the NVSim behavioural simulator for the energy, delay and area modeling of memories was motivated by its wide adoption in literature, availability and ease of use.

The NVSim model for our selected STT-MRAM cell is based on the perpendicular tunnel magnetoresistance device (P-TMR) presented in [30]. The cell access type is planar CMOS-based and the bit-cell stack is 1T-1MTJ. The

Table 5.1: Energy and access latencies for SRAM, ReRAM and MRAM. Technology considerations: HP tranistors at 32nm node.

Parameters	**SRAM**	**OxRAM**	**MRAM**
Read energy	0.051nJ	0.032nJ	0.037nJ
Write energy	0.049nJ	3.702nJ	0.729nJ
Read latency	0.232ns	0.757ns	1.587ns
Write latency	0.176ns	20.083ns	10.261ns
Leakage	121.259mW	48.501mW	31.095mW
Area	$0.315mm^2$	$0.203mm^2$	$0.167mm^2$

standard supply voltage is 1V (with full V_{dd} across the BL to drive the switching of the STT-MRAM cell) and the interface is SRAM compatible. Depending on the gate oxide thickness the max voltage (overdrive) can extend upto 1.5V. We will require at-least 1.3V on the access transistor gate to generate the current required to switch the MTJ ($\sim$150μA). The memory cell size is 0.1284μm^2. The OxRAM cell is based on 4Mb embedded SLC resistive-RAM demonstrated in [189]. The bit-cell for OxRAM is 1T-1R with an I/O access transistor (with a drive voltage across the BL of about 1.5V) and the interface is SRAM compatible. The periphery requires 1.8V whereas the cell array requires 3.3V. The access transistor sizing depends on the NVM parameters like Read, Set and Reset voltage and is critical to determine the pitch and macro design. For both the STT-MRAM and ReRAM, the bit-cell area and resistance is dominated by the selector. It is to be noted that for smaller memory/cache sizes, similar to CACTI, NVSim numbers are quite pessimistic as a result of the abstraction level of the simulators and do not take into account the circuit modifications required to optimize a design for a particular NVM technology. It must also be mentioned again that like in the previous chapters, these numbers were representative of the NVM technology maturity at the time and do not represent state of the art highlighted in table 2.2 (Chapter 2).

Figures 5.2 and 5.3 show how some of these memory/cache parameters scale, relative to the SRAM parameters, with increasing memory sizes according to NVSim. Evidently, a trend towards better numbers for the NVM based memories with increasing sizes can generally be expected (although realistically not to the extent as shown in the figures). Thus, it is quite safe to claim that our results and explorations are for fairly pessimistic scenarios and can will most certainly yield better returns with improved models and approximations. As can be seen, leakage for both NVMs is smaller than for SRAM, being almost half in the case of STT-MRAM. Considering their better scaling properties and relatively low access latencies, both technologies seem to be promising replacements of the SRAM for the higher levels of the memory hierarchy.

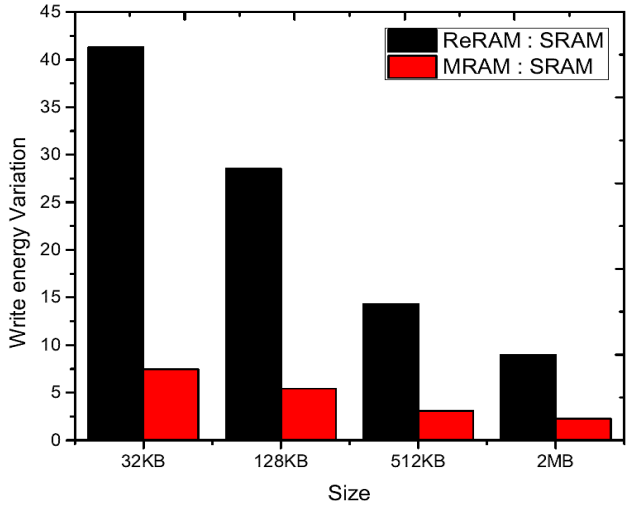


Figure 5.2: The variation in the ratio between the write energy of NVM and SRAM with increasing sizes.

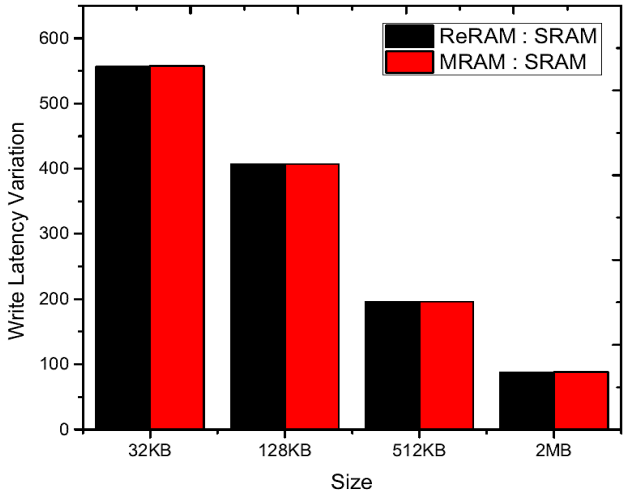


Figure 5.3: The variation in the ratio between the write latency of NVM and SRAM with increasing sizes.

5.3.1 STT-MRAM vs ReRAM

While we initially considered ReRAM (HfO_2 based OxRAM specifically) as our choice of NVM for low/ultra-low power embedded solutions at the highest levels of the memory, a number of factors have motivated the switch to STT-MRAM as the NVM of choice for our preferred exploration space.

- Endurance: A part from the stringent performance requirements, endurance is one of the biggest factors that is crucial for a memory technology to be considered as a viable option for the highest levels of the memory hierarchy. It is here that ReRAM pales in comparison to STT-MRAM. While, the best experimentally reported ReRAM endurance numbers hover around 10^{10} cycles (for OxRAM) ([79, 41]), STT-MRAM has repeatedly shown endurance around and above 10^{16} cycles [5, 105]. This comparison has been clearly outlined in Table 2.2 (chapter 2).

- Switching voltage: Apart from the low endurance, the high voltage required to switch the cell in ReRAM and the physics behind the formation of conduction paths (a destructive process) in ReRAM fundamentally limits it. After years of research, the lowest stable RESET switching for any ReRAM is around 1.2V (Compared to 0.4V for STT-MRAM). This is impractical, especially for the deeply scaled nodes. This also leads to a problem when considering an appropriate selector, since I/O transistors required to drive this voltage will effectively kill all the area benefits of NVMs when compared to SRAM. STT-MRAM switching voltages on the other hand can be reached by means of the simple planar CMOS transistors and/or FinFET technology for the deeply scaled nodes (N14 onwards).

- Variability: One of the aspects that is currently holding back the entry of ReRAM into the market even when it comes to SCMs, is it's intrinsic variability (also including read noise and program noise). The resistance drift (gradual drop in the resistance values of the SET and RESET state with time) phenomenon is also a major factor contributing to this variability in ReRAM. All these factors are magnified since the conducting filament is believed to approach a few atoms in size. Program and read noise are expected to increase with lower programming current, due to: (1) the reduction in the size of the conducting filament and (2) the increased relative effect of defects on conductivity of the filament [3]. Since, high-density applications will require the cell to function with low programming currents (10-50μA), high variability and noise are significant product-level concerns. The read margin for OxRAM is $\sim$200nA, and $\sim$2μA for CBRAM (for a pulse of 300ns, difference between

LRS and HRS read currents at 3 sigma) [17]. The retention BER for low compliance current $\sim 50\mu$A is $\sim 10^{-4}$ for CBRAM and $\sim 10^{-2}$ for OxRAM [17]. On the other hand, WER of $\sim 10^{-7}$ has been shown for 11nm MTJ devices at switching current of 7.5μA (10ns pulse) [151].

Due to the multitude of reasons listed above, we have decided to consider STT-MRAM as our NVM of choice for the rest of the chapters since the focus of the thesis is on the highest levels of the memory hierarchy. Here, the write frequency simply overwhelms the ReRAM technology and the problem is compounded for data caches.

5.3.2 NVM drop-in

One of the main problems with these resistive memories is their asymmetry in the latencies of the read and write operations. As shown in table 5.1, write latencies are significantly larger than their read counterparts. A simple simulation can show that this asymmetry has a major impact on performance when NVMs are used on the first levels of the memory hierarchy, even for instruction caches with low write frequencies. For instance, we have analyzed the impact of a NVM based I-Cache on the performance of a Single Core ARM processor with 32KB IL1 cache, 64KB DL1 cache and a 2MB unified L2 cache. We have used the microarchitecture simulator Gem5[15] for this analysis, assuming that we can maintain the read access time of the SRAM cache (same cycles) while write operations are assumed to range from 10 to 20 cycles (where 1 cache cycle = 2 global/processor clock cycles (1ns) for the 1GHz ARM processor in our GEM5 framework), according to the NVMs write latencies assuming a processor frequency below 1 GHz.

Figure 5.4 shows the performance penalty of the system on replacing just the SRAM instruction cache by a NVM counterpart with similar characteristics (size, associativity...). The Data cache and the unified L2 remain SRAM based. We ran the simulations with a subset of the SPEC CPU2006, MediaBench and the DSPStone benchmarks. For any write latency considered, we observe a clear and unacceptably large performance overhead compared with the baseline. Indeed, *djpeg* may suffer up to 40% performance penalty if the NVM based instruction cache write latency is as high as 20 cycles. The main conclusion of this analysis is that although STT-MRAM memories are potentially good candidates to replace SRAM caches, an immediate replacement (simple drop-in) is not advisable and some architectural modifications are required to reduce the impact of their large write latencies.

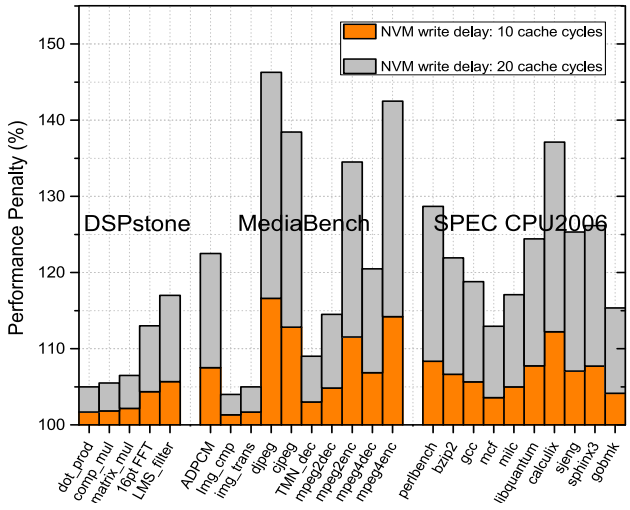


Figure 5.4: Performance penalty for the NVM I-Cache, relative to SRAM I-Cache, considering two different ratios of write vs read latencies (delay). SRAM Icachebaseline = 100%.

5.4 Extended MSHR to address NVM limitations

Lockup-free caches are fairly common on modern processors. The first organization proposed for such a non-blocking cache was given by Kroft [104] by means of a set of registers called MSHRs (Miss Status Holding/Handling Registers). These registers typically hold enough information on all outstanding misses to complete the load process like

- the physical address of the block
- which word in the block
- destination register number (if data)
- mechanism to merge requests to the same block
- mechanism to ensure accesses to the same location execute in program order

Primary and secondary misses can use the same MSHR. The organization of a simple MSHR is shown in Figure 5.6. Whenever a cache miss occurs, a free entry is looked up in the MSHR, and its valid bit (V) is set to 1. If there is no

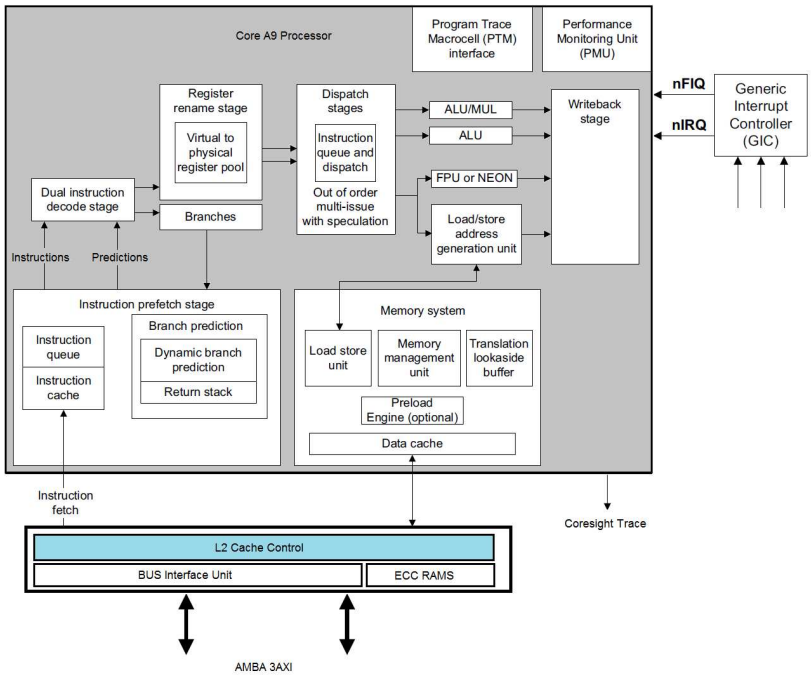


Figure 5.5: The traditional memory hierarchy in a single core ARM platform (Cortex A9 here) [6].

free entry left, the cache gets blocked and the processor stalls. Otherwise, the memory block of the missing request is annotated in the *block address* field so it is possible to track further misses to the same block and execution proceeds. Whenever the miss is served from lower cache levels, the requested word is handled by the processor and the cache is conveniently updated. The valid bit (V) is reset to make the entry become free again.

As stated earlier our focus is on the Instruction cache of single Core ARM like processors having a traditional memory hierarchy (specifically ARM Cortex A9 in our case since, it is most closely mimicked by the GEM5 simulator). The cache structure includes a L1 Instruction Cache, L1 Data Cache and a unified L2 Cache (Figure 5.5). The original MSHR organization that is taken as the base for our explorations is detailed in Figure 5.6. This baseline SRAM based instruction cache organization will be termed as BS-Icache from here on. The instruction cache here is a classic 2 way associative cache. It has a single ended port (Read/Write) and is divided into 4 banks (configurable). Usually, cached memory reads/writes that match a particular cache tag (with Valid, Read and

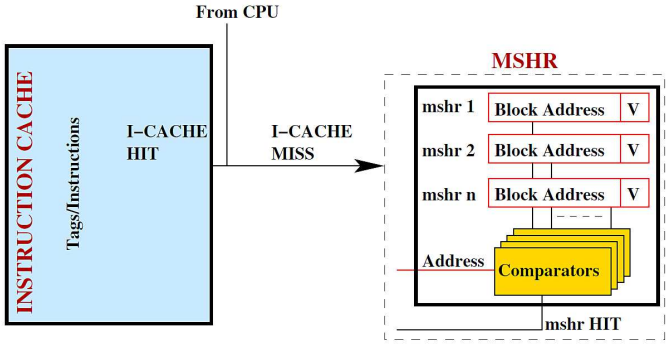


Figure 5.6: The original MSHR based cache organization

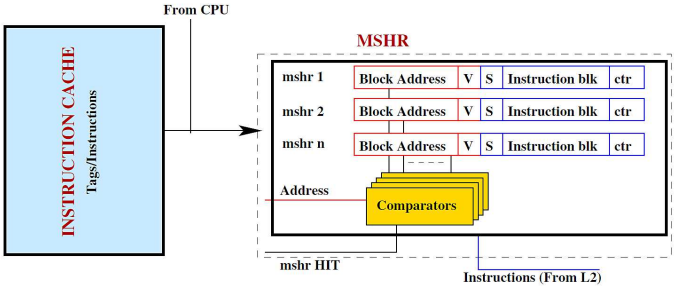


Figure 5.7: The modified MSHR (EMSHR) based cache organization

Write flags) are completed upon the appropriate 'CACHE-HIT' interval. If this is not the case, then it is termed as a 'CACHE-MISS' and the request is forwarded to the Miss Status Handling Registers [104] [47]. The cache structure, along with the terminology used for references is highlighted in Figure 5.8

If we are to make the cache become NVM-based, there is a clear need to filter write operations to NVM as motivated in previous section. Moreover, it is also advisable to smartly schedule the NVM cache updates to hide as much latency as possible. In this chapter, we analyze the feasibility of using NVMs for the L1 I-cache, where the only source of write operations are the L2 updates triggered by L1 I-cache misses. We propose using a modified version of Kroft's MSHR organization, in which a small intermediate buffer is used to hold instructions arriving from L2 before they actually update the instruction cache. This *extended/enhanced* MSHR (EMSHR) is detailed in Figure 5.7. Every register in the EMSHR is extended to hold the complete IL1 block (block size = 128 Bits) corresponding to the memory request it is tracking. In addition, a

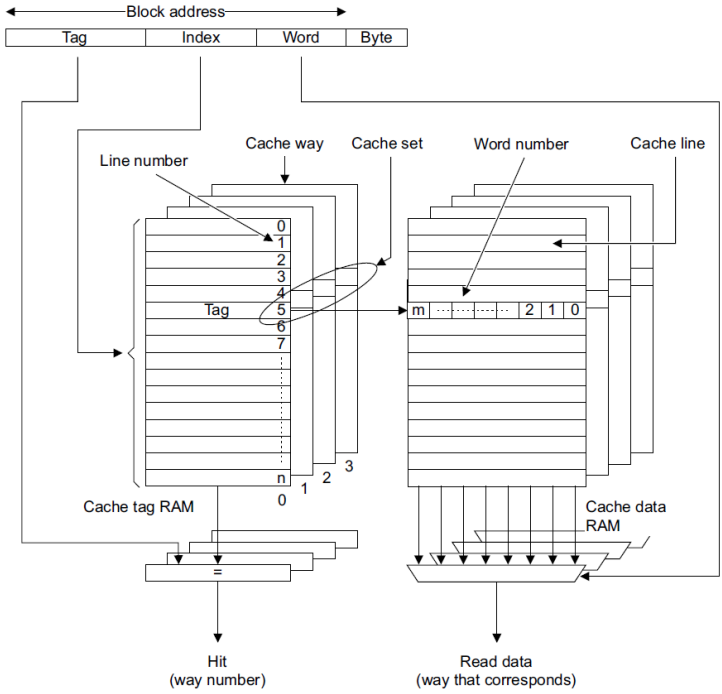


Figure 5.8: The cache structure and associated terminology [6].

new control bit 'S' is added to signal that the entry holds valid instructions, i.e. the cache miss was already served and the memory block has been stored in the EMSHR instruction block. Notice that we do not necessarily need to start the transfer of the memory block from the EMSHR instruction block to the NVM IL1 immediately after its arrival from the L2; we can hold it in the EMSHR instruction block to see if there is some reuse. The EMSHR thus effectively acts like a very small (up to 4Kbit: 32 instruction blocks) fully associative cache between the NVM IL1 and the L2 cache. The architecture of the cache also needs to change a little, as the data path now includes the EMSHR. A multiplexer/selector must be included in the modified architecture to allow the processor to select whether to fetch instructions from the STT-MRAM array or the EMSHR.

The fetch sequence is now slightly different when using the extended/enhanced MSHR:

1. When the CPU issues a new fetch, the NVM instruction cache tag array

and the block tag corresponding to each entry in the EMSHR are searched to look for a hit. If the access hits any of the two structures, the required instruction is served within that access cycle. An EMSHR access can only be considered a hit if the matching entry has its 'S' bit set. This way, only one of the two structures would hold the required instruction (i.e. there might not be any duplicated hit).

2. If the EMSHR associative search was successful, but the 'S' bit was reset, the bit 'V' (*validity check*) indicates if there was a previous miss to the same block. If that is the case, this *secondary* miss is annotated in the EMSHR and the execution proceeds.

3. Otherwise, a *primary* miss has occurred and a free EMSHR must be allocated. Then, the valid bit is set and block request address of the free EMSHR is recorded. The cache itself does not stall the processor, but if it was an instruction fetch (no prefetching request) from a non-multithreaded processor, it would normally lead to a stall condition.

4. Finally, once the miss is served from next lower level, the instruction block is written into the allocated EMSHR and the 'S' bit is set. Subsequent accesses to this memory block will hit this EMSHR.

There are some other salient points to be taken into consideration in order to complete the picture of our proposal:

- If during step 3 there was no free EMSHR (i.e. all MSHRs have V=1), but there is at least one entry with S=1 (i.e. it is holding valid instructions) a replacement policy must take place. Since we restrict the proposal to instruction cache, there is no write-back involved (since no IL1 line can be dirty), but we still need to select one MSHR to track the new miss from all the candidates (i.e. all MSHRs with V=0 and S=1). We have implemented the 'Least Recently Used' (LRU) policy.

- On the other hand if all MSHRs are in the V=1 and S=0 state (i.e all waiting for pending misses) then the processor stalls.

- A *promotion* mechanism must be implemented so that EMSHR entries (occasionally) update the main NVM instruction cache. One of our goals is to minimize NVM writes to alleviate performance and endurance limitations. Therefore, we must carefully select which blocks are actually written to the NVM array. We propose to *promote* a EMSHR instruction block to the NVM array whenever it has been accessed more than a fix (parametrized) number of times (or a 'threshold'). Thus, a counter is associated with each EMSHR to track the number of references to that

block. The rationale behind this proposal is to use the NVM array mainly for loop codes (that allows the code to be executed repeatedly), while mostly sequential code only resides in EMSHRs.

Regarding performance, the proposed mechanism decouples the miss from the NVM updating thus removing the long latency write operation from the critical path. However, those writes may eventually happen and the processor may try to fetch a new instruction while a *promotion* is taken place (since the *promotion* may take as long as 20 cycles). Given the banked nature of the NVM array, there will be no conflict if both operations target different banks. Otherwise, the processor fetch must be stalled and the system performance will slightly degrade.

In the next section we will show how this mechanism drastically reduces performance penalty while keeping the benefits of NVM memories (heavily reduced leakage/retention property and compact area). An exploration of the existing knobs (namely the number of MSHRs, the *promotion* threshold and the NVM array size) is performed, outlining the performance and energy trends of each of them.

5.5 Exploration and Analysis

In order to evaluate the effectiveness of the proposed modifications and mechanisms, we implement them via the GEM5 simulator ([15]). The gem5 simulator is a modular platform for computer-system architecture research, encompassing system-level architecture as well as processor microarchitecture. Listed below are some of the key features of Gem5:

- *Multiple interchangeable CPU models*: gem5 provides four interpretation-based CPU models: a simple one-CPI CPU; a detailed model of an in-order CPU, and a detailed model of an out-of-order CPU. These CPU models use a common high-level ISA description. In addition, gem5 features a KVM-based CPU that uses virtualisation to accelerate simulation.
- *A NoMali GPU model*: gem5 comes with an integrated NoMali GPU model that is compatible with the Linux and Android GPU driver stack, and thus removes the need for software rendering. The NoMali GPU does not produce any output, but ensures that CPU-centric experiments produce representative results.
- *Event-driven memory system*: gem5 features a detailed, event-driven memory system including caches, crossbars, snoop filters, and a fast

and accurate DRAM controller model, for capturing the impact of current and emerging memories, e.g. LPDDR3/4/5, DDR3/4, GDDR5, HBM1/2/3, HMC, WideIO1/2. The components can be arranged flexibly, e.g., to model complex multi-level non-uniform cache hierarchies with heterogeneous memories.

- A *trace-based CPU model* that plays back elastic traces, which are dependency and timing annotated traces generated by a probe attached to the out-of-order CPU model. The focus of the Trace CPU model is to achieve memory-system (cache-hierarchy, interconnects and main memory) performance exploration in a fast and reasonably accurate way instead of using the detailed CPU model.

- *Homogeneous and heterogeneous multi-core*: The CPU models and caches can be combined in arbitrary topologies, creating homogeneous, and heterogeneous multi-core systems. A MOESI snooping cache coherence protocol keeps the caches coherent.

- *Multiple ISA support*: gem5 decouples ISA semantics from its CPU models, enabling effective support of multiple ISAs. Currently gem5 supports the Alpha, ARM, SPARC, MIPS, POWER, RISC-V and x86 ISAs. See Supported Architectures for more information.

- *Full-system capability*: gem5 can simulate a complete system which includes providing an operating system based simulation environment.

- *Multi-system capability*: Multiple systems can be instantiated within a single simulation process. In conjunction with full-system modeling, this feature allows simulation of entire client-server networks.

- *Power and energy modeling*: gem5's objects are arranged in OS-visible power and clock domains, enabling a range of experiments in power- and energy-efficiency. With out-of-the-box support for OS-controller Dynamic Voltage and Frequency (DVFS) scaling, gem5 provides a complete platform for research in future energy-efficient systems. See how to run your own DVFS experiments.

- *Co-simulation with SystemC*: gem5 can be included in a SystemC simulation, effectively running as a thread inside the SystemC event kernel, and keeping the events and timelines synchronized between the two worlds. This functionality enables the gem5 components to interoperate with a wide range of System on Chip (SoC) component models, such as interconnects, devices and accelerators. A wrapper for SystemC Transaction Level Modelling (TLM) is provided.

The use of the GEM5 architectural simulator was motivated by its wide adoption in literature, availability, accuracy and and our exploration space (cache architecture in general purpose RISC machine). We run all our experiments with the *arm_detailed* CPU type which mimics an ARM Cortex A processor with a clock frequency of 1GHz. We run all our simulations in the System Call Emulation (SE) Mode. In this mode, one only needs to specify the binary file to be simulated. This binary file should be statically compiled since the simulator does not dynamically link executables. We opt for this mode to reduce simulation time and the complexity involved. We keep the default values for cache configuration: 32KB 2-way associative L1 instruction cache, 64KB 2-way associative data cache and a 1MB 16-way associative unified L2 cache. The instruction cache originally includes 2 MSHRs whose functionality have been extended as explained in previous section. A single I-cache block is 4 instructions wide (128 bits), so each register in the EMSHR must be able to hold 4 instructions.

For simulations, we include a wide range of representative benchmarks from the DSPstone, MediaBench and SPEC CPU2006 benchmark suites. Since the ARM platform is used for our explorations (via GEM5), the benchmarks are compiled by means of the GNU ARM embedded toolchain (arm-linux-gnueabi host). Compile time optimizations are limited to level 1 so as to study the effect of optimizations (carried out later in 6.5). We opted to limit input datasets for applications at the minimum to aid quicker simulation. For eg, we utilized the 'DMINI DATASET' option while compiling PolyBench benchmarks (one of the reasons for very small performance penalty). The inputs for Mediabench were comparatively larger (relative to SPEC and PolyBench) and thus resulted in a larger performance penalty (HD images etc). The performance penalty and energy reduction trends with architectural modifications alludes to this since SPEC benchmarks are more irregular. The benchmarks were chosen from each suite to be more representative of varied types and not repetitive. The Mediabench benchmarks include image, audio and video applications. The SPEC benchmarks include both floating point and integer applications.

The *promotion threshold* introduced in previous section (i.e. the number of references to a busy EMSHR before it is transferred to the NVM I-cache) results in a crude separation of the code into loop dominated content and otherwise in the modified cache organization. Since, the number of EMSHRs is limited and they cannot possibly hold the entire non-loop content of the code, evictions are regularly present that result in increased instruction accesses from L2. But overall, improved performance benefits can be attained. The performance penalty reduction due to the extended MSHR is given in Figure 5.9. It is fully clear that our modifications to the instruction cache architecture helps in reducing the performance penalty significantly, to less than 5% in all

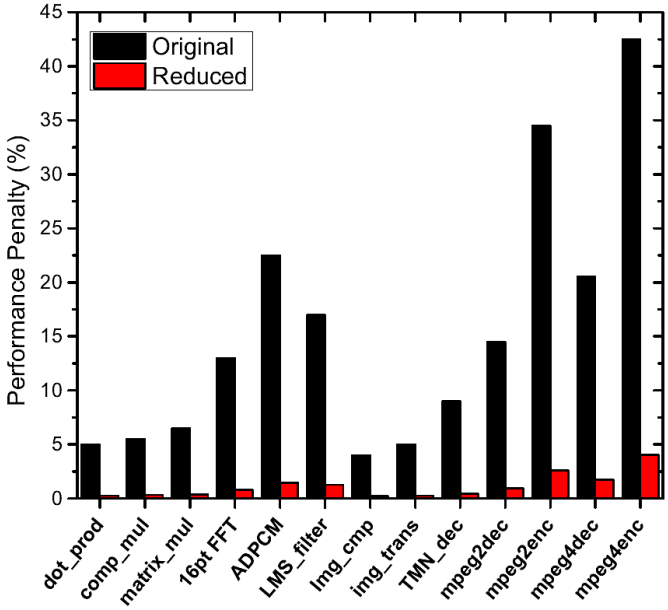


Figure 5.9: Performance Penalty reduction with changes in architecture (EMSHR introduction and associated changes).

benchmarks. Expect for the few benchmarks with a seemingly lower read:write ratio, like the mpeg decoders, mpeg encoders and ADPCM among the larger ones and 16point FFT and LMS filter among the smaller ones, the performance penalty is all but negligible.

To evaluate the impact of our proposal on performance more acutely, we explored two key EMSHR parameters: promotion threshold and EMSHR size (based on the number of registers). The fact that there exists a small but significant performance penalty for the largest benchmarks even after the initial architectural modifications makes these explorations more worthwhile to pursue. Figure 5.10, details the performance penalty variation on the tuning the the two key MSHR parameters in our modified STT-MRAM based L1-cache organization. From figure 5.4 (section 5.3), we know that the performance penalty due to a simple NVM I-cache can reach up to 45%. This can be reduced to 1% penalty with a 32 entry EMSHR (4Kbit size) and a promotion threshold of 12.

We see that performance penalty generally reduces for larger EMSHR sizes and higher promotion thresholds. Increasing the EMSHR size allows more

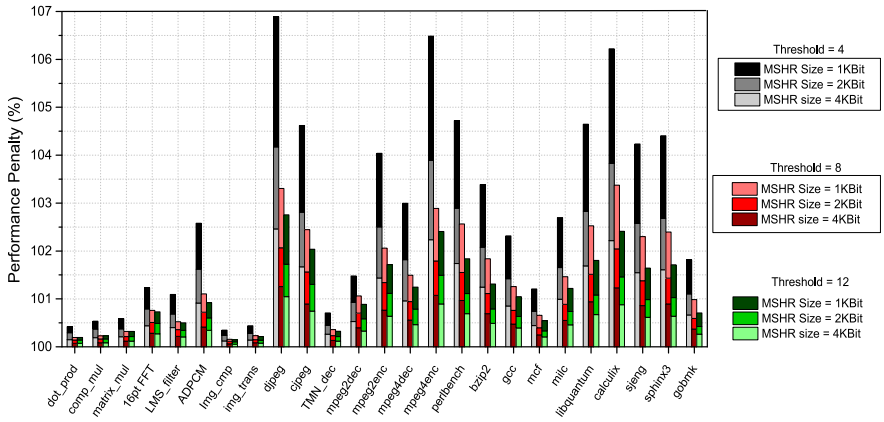


Figure 5.10: Performance penalty changes of the STT-MRAM based modified I-cache (relative to the baseline SRAM IMO performance of 100%) for different EMSHR variations.

instruction blocks to be read frequently without accessing the L2 often and thus avoids unnecessary evictions. A reduction in the performance penalty is expected in such a scenario. However, there are physical routing limitations on the extent to which the EMSHR size can be increased and so we limit it to a few KiloBits. The area overhead and tag search will also start to become an issue. Increasing the EMSHR promotion threshold means lesser instructions blocks are written into the STT-MRAM Icache (only the most reused loop bodies are promoted). In most of the bigger benchmarks, the percentage of IL1 capacity misses is very significant, so most code blocks will be evicted before they are fully reused. Given the large write penalty, its better to avoid those writes unless the amount of immediate reuse is large enough even if larger thresholds lead to an increase in L2 accesses. From the figure, it is abundantly clear that our modifications to the I-cache organization helps in reducing performance penalty significantly, to less than 4% for all benchmarks even with a EMSHR capacity of just 8 entries (i.e. 1Kbit).

We also explore the energy consumption variation of the modified system with STT-MRAM I-cache with changes in EMSHR parameters (Figure 5.11). The energy numbers are obtained via NVSim (Table 5.1) and include the dynamic and static energy consumption of the L1 I-cache and EMSHR. The read energy consumed on L2 accesses due to L1 I-cache misses is also included here. This is important since, the modification to the L1 I-cache organization significantly affects this (leads to increased L2 instruction accesses). For all the EMSHR sizes explored, as expected, the larger the EMSHR, lower has been the energy

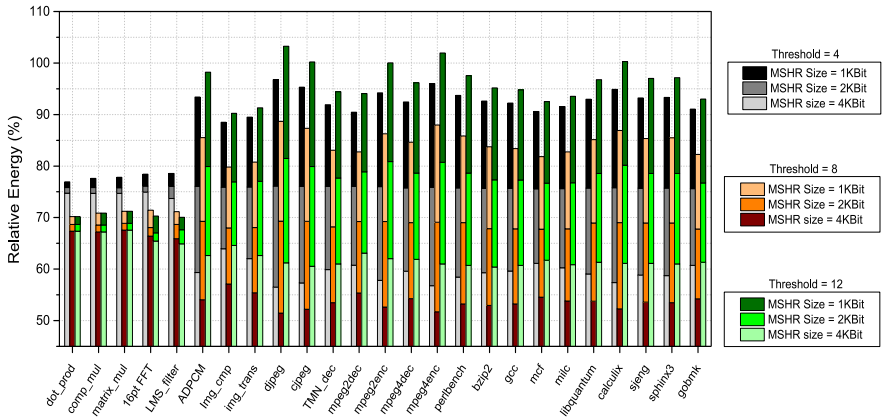


Figure 5.11: Energy comparison for different MSHR variations.

consumption. Thus, if we increase the number of registers, we reduce the number of energy expensive NVM writes without incurring extra L2 accesses.

The promotion threshold exploration shows a more interesting behaviour. On increasing the threshold from 4 to 8 accesses, the total energy consumption of all benchmarks reduces. As always, filtering writes to the NVM is globally more energy efficient even at the cost of extra L2 accesses. However, upon further increasing the threshold up to 12 accesses, the largest benchmarks show a clear increase in the energy consumption. Not promoting the loop codes soon enough (and sometimes never if the loop iterations are below 12!) leads to an important increase in the L1 misses. As a consequence, the number of accesses to the bigger L2 increases. Hence, an intricate balance exists between the NVM write energy consumption and the energy consumed by accessing the L2 more often. The small size of the instruction code for the DSPstone benchmarks make these explorations (both performance and energy) particularly favourable for them compared to the others, since most of the instructions can fit in the EMSHR and there is minimal use of the STT-MRAM array.

To further study the impact of our architectural modifications on the energy consumption of the L1 I-cache organization, we compute and compare the energy breakdown of the STT-MRAM based modified I-Cache organization and the original SRAM based I-cache organization (Figure 5.12). We fix the MSHR promotion threshold and MSHR size to 4 (accesses) and 2Kbit respectively in these simulations. The relative energy consumption for the 2 different scenarios (STT-MRAM based I-cache and SRAM based I-cache) are represented by their respective cumulative bars for each benchmark. The energy consumption of SRAM-based architecture is taken as the baseline here. These bars detail the

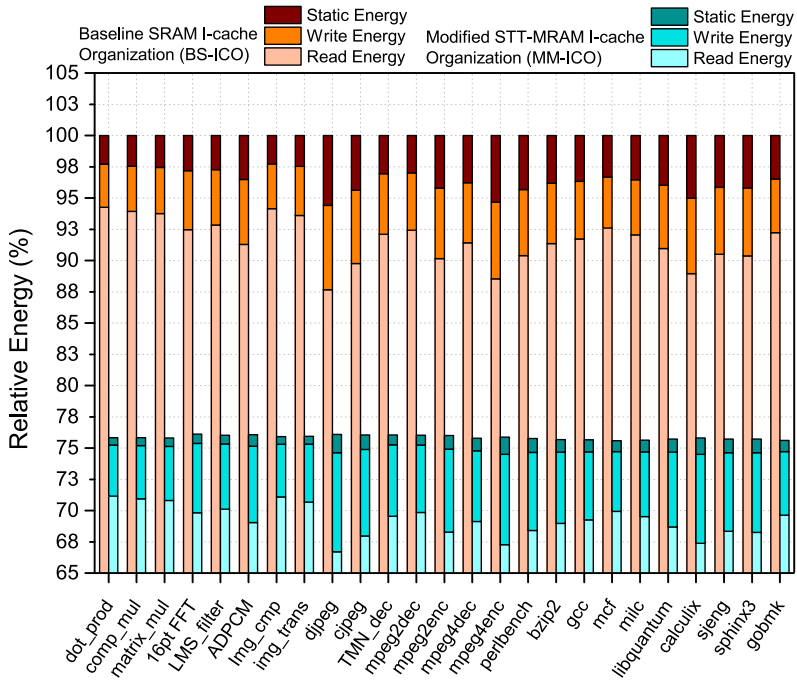


Figure 5.12: Energy consumption breakdown of the modified STT-MRAM and original SRAM based L1 I-cache organizations (MSHR size = 2Kbit, threshold = 4). All numbers are relative to the original SRAM based I-cache organization (100%).

read, write and static (leakage) energy contributions. Energy contributions from both the L1 I-cache and EMSHR are included. Again, the read energy consumed by means of L2 accesses due to L1 I-cache misses is included.

From the energy breakdown, It is quite clear that the L1 I-cache read energy dominates the total energy consumption in the instruction memory organization. The energy consumption trends here can be explained by means of 3 different attributes.

- **The profile of the application benchmarks**: Like it has been pointed out before, some of the applications have a large read:write ratio (loop-dominated) and some others have a comparatively lesser read:write ratio. Looking closely at some of the larger benchmarks (like the mcf, milc, image transcoder, image comparer, mpeg-2 and 4 decoders and encoders, ADPCM and TMN-decoder) this distinction becomes clearer. The image

comparer and transcoders, mcf and milc etc have a much larger read:write ratio as compared to the other benchmarks. This difference can clearly be seen in the NVM based instruction cache organizations due to the much larger write energy (relative to SRAM) and hence larger visible difference. The large read;write ratio is most likely due to the loop dominated instruction code.

- **The architectural modifications**: Despite the huge difference in the write energy between the NVM based I-caches (∼10 for MRAM) and the SRAM based I-cache, we can state that the energy consumption of the NVM based instruction cache organizations remain fairly lower relative to the SRAM based instruction cache organization (achieving, on average, an 24% reduction for the modified configuration). Appropriate modifications and adjustments to the tunable knobs/parameters like MSHR size and *promotion threshold* can also possibly lead to even lower energy consumption for the MRAM based I-cache organization relative to the SRAM based one (in-case of certain benchmarks). The read:write ratio for the benchmarks isn't enough to explain this. The architectural modifications to the NVM based caches ensure that most of the writes are to the small modified MSHR buffer. This ensures that the total number of writes to the NVM I-caches is minimal and the resulting the impact of their much larger write energies is greatly reduced.
- **The energy contribution of the components**: The read energy per access for the SRAM and MRAM based I-caches are outlined in Table 5.1 (Section 5.3). This combined with the above points paints a clear picture of the energy consumption breakdown. Another important thing to note here is the leakage contribution. For the SRAM based instruction cache organization, the leakage contribution almost equals that of the write energy contribution in the largest benchmarks ('*mpeg4 encoder, djpeg*' etc). As we move towards lower technology nodes, this contribution will obviously increase significantly (see Section 5.3). This further motivates the transition to NVM based memories for the high level instruction memory organization.

Area Explorations

In all the above experiments we have considered a 32KB STT-MRAM, which is the same size as that of the base line SRAM I-cache. The STT-MRAM I-cache is smaller in area when compared to the SRAM I-cache ($0.167mm^2$ vs $0.315mm^2$) due to the much smaller cell size ($14mm^2$ vs $146mm^2$) even when we take into account the larger transistor to drive higher write current. Hence,

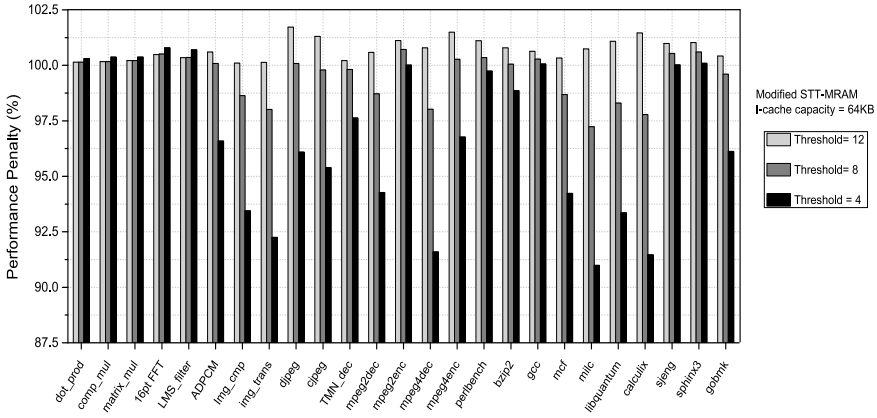


Figure 5.13: Performance Penalty variation of the modified STT-MRAM I-cache for different promotion thresholds on increasing the capacity to 64KB. All numbers are relative to the original SRAM based I-cache organization (100%).

we explored the utilization of larger capacity caches. For these experiments we fixed the EMSHR size at 2Kbit. The capacity of the cache be realistically increased by a factor of 2 (64KB STTMRAM = $0.294mm^2$), taking into account the area overhead of the 2 KBit EMSHR ($0.020mm^2$), and the total area of the proposed memory architecture will remain comparable to the baseline SRAM I-cache area. Figure 5.13 shows the significant performance improvements achieved with this larger (64KB) STT-MRAM based modified I-cache. There are two important aspects to be highlighted here: firstly, the performance penalty is completely removed and even some minor improvements may be observed. Secondly, with this configuration, the lower the threshold, the better the performance (just the opposite of the behaviour observed in figure 5.10). The huge reduction in capacity misses makes it much more convenient to promote instruction blocks sooner since they are likely to reside in IL1 much longer than before.

Figure 5.14 gives the changes in the relative energy associated with this larger (64KB) STT-MRAM based modified I-cache relative to the original SRAM baseline. It's quite clear from this figure and the earlier energy comparisons that a larger cache is clearly beneficial for the larger benchmarks (most of SPEC CPU2006 and Mediabench) and especially those with loop dominated codes provided that the loop iterations are sufficiently large. There is an important reduction in both, the number of writes to the STT-MRAM array and the number of L2 accesses, specially when the promotion threshold gets lower. Only for the smallest benchmarks, where the capacity misses were not a problem,

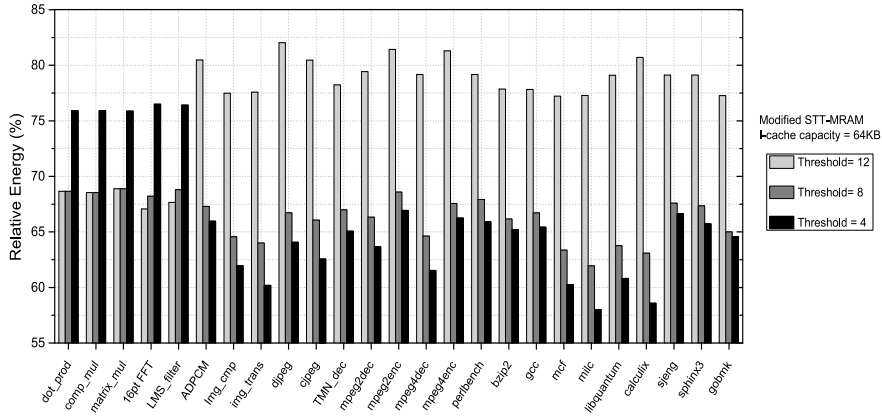


Figure 5.14: Energy variation of the modified STT-MRAM I-cache for different promotion thresholds on increasing the capacity to 64KB. All numbers are relative to the original SRAM based I-cache organization (100%).

decreasing the threshold does not lead to energy improvements.

Performance penalty and energy consumption are not the only factors that hinder NVM based caches. Endurance plays a huge part in the feasibility of these proposed caches and research is being extensively carried out to improve the life-time of these new NVM cells. One of the more important aspects of the modification to the instruction cache that has been proposed here is the reduction of writes to the I-cache. This will most certainly have a significant bearing on the endurance and lifetime of the cache. In addition to the architectural modifications, it would also be interesting to explore the influence of changing cache size on the endurance. Endurance here can roughly be gauged by the write reduction (%) of the NVM based I-cache. Figure 5.15 shows the reduction in the number of writes to the NVM based cache on changing the traditional instruction cache architecture via the MSHR enhancements for different instruction cache sizes. Here a 70% reduction hints that there are only 30 writes into the I-cache for the Modified architecture compared to 100 writes for the original architecture.

The significant amount of write-reduction here shows that the proposed architectural modifications also help improve the lifetime of the NVM based cache, thus making it more worthwhile to pursue in the future. It can also be noticed that the reduction of the cache size negatively affects this aspect for larger benchmarks. This hints at the fact that a good portion of the misses to the instruction cache in the case of the larger benchmarks are capacity misses.

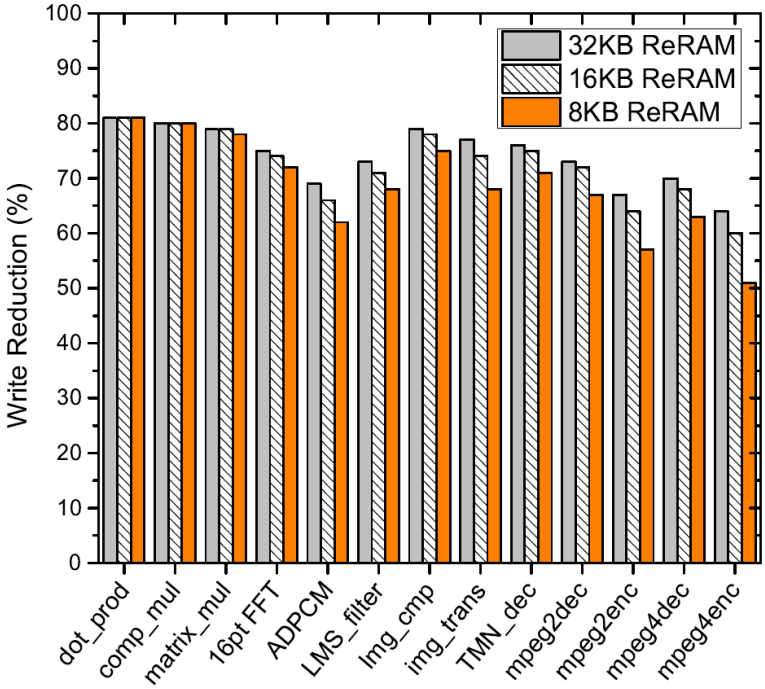


Figure 5.15: The variation in the write reduction for selective sizing of NVM cache.

5.6 Conclusion

This chapter presents a modification of the traditional MSHR organization for the effective exploitation of STT-MRAM based L1 I-caches within the context of high performance general purpose systems. Our exploration shows that the appropriate tuning of selective architecture parameters can reduce the performance penalty introduced by the NVM to extremely tolerable levels ($\sim$1%). In addition, we show that we can also obtain significant reductions in energy consumption ($\sim$35%), given that the NVM technology has a favourable read energy per access when compared to SRAM. However, the pareto-optimum values for the different parameters are heavily application and platform dependent. Furthermore, the use of NVMs also allows gains in area and this in turn can be utilized to accommodate I-caches with more capacity ($\sim$2-3 times for STT-RAM). When configuring the modified NVM based I-cache to occupy the same area as the original SRAM-based configuration, the NVM based system outperforms the SRAM baseline and leads to even more energy

savings. Finally, we also show that STT-MRAM memories are potentially good candidates to build feasible L1 I-caches even for relatively conservative scenarios.

Chapter 6

STT-MRAM based D-cache explorations for High Performance General Purpose Processors

6.1 Introduction

In the previous chapter, we highlighted the challenges associated with the substitution of a traditional SRAM based instruction cache organization with an NVM (STT-MRAM specifically) for an ARM like general purpose platform. With increasing amount of data processing nowadays, reducing the energy consumption of the D-cache organization has become an important part of the reduction in the system's overall energy consumption. Like before, we only focus on STT-MRAM as our NVM of choice with our context in mind and explore the replacement of the traditional SRAM based L1 D-cache (DL1) by a STT-MRAM based one.

Despite the low energy, leakage and very good endurance, STT-MRAM latency is an issue when we are targeting higher level memories. Previous concerns about STT-MRAM and other similar NVM technologies were along the lines of write-related issues. The read:write latency depends a lot on the R-Ratio (TMR in the case of STT-MRAM) in these NVM technologies. With the maturation of the STT-MRAM technology it has become clearer that a very high R-Ratio

(more than 200%) is not realistic, at-least currently, taking into account the cell stability and endurance ([151]). Hence, the read latency has become the new major bottleneck to overcome for the substitution of SRAM by STT-MRAM, particularly at the L1 level of the memory hierarchy. Due to this, there has been a shift from a 1T-1MTJ to a 2T-2MTJ complimentary type bit-cell([5],[147]) to sense the differential load over the bit-line quicker and read quickly. However, the read energy is expected to be higher in this case, but we are limited by selective studies that do not show relevant data/focus on both read and write access.

As a result of the above mentioned latency issues, a direct drop-in replacement of SRAM by STT-MRAM in the D-caches organization is not feasible. We propose a novel D-cache configuration that is able to overcome the read limitations of the STT-MRAM by means of an intermediate buffer that is termed as the 'Very Wide Buffer' (VWB). This is a significant extension of the design methodology initially proposed in [173] and [102]. We extend the design processes and architectural modifications outlined in both scenarios and come to a solution optimized for our particular specifications. These architectural changes along with appropriate data allocations schemes coupled with code transformations and optimizations are best utilized to make our proposal viable. Since the focus is on system level changes, we will not delve deeply into technology and circuit related matters. To the best of our knowledge, this is the first such work on NVM based L1 D-cache organizations that manages to specifically tackle the read latency limits via micro-architectural modifications and code transformations. This chapter will illustrate this on single core ARM like general purpose platforms, but the principles are not limited to this type of platform.

The chapter is organized as follows: Section 6.2 discusses similar work and recent developments with respect to the use of NVMs. Section 6.3 motivates the need for architectural modifications in order to replace the SRAM based D-cache by a NVM counterpart. Section 6.4 details the proposed architectural modifications to deal with the performance penalty raised by the NVM's read latency limits. Section 6.5 lists the code transformations and optimizations that enable us to fully exploit our architectural modifications and avail the benefits of the NVM cache. Section 6.6 presents the results of the proposed methodologies, and an exploration of the effects of the different tune-able parameters along with certain comparisons. Section 6.7 concludes the chapter.

6.2 Related Work

There have been a number of proposals based on hybrid NVM/SRAM organizations for different levels of the memory hierarchy. Like it has been stated in section 5.2, there is very few work on replacing the highest levels of the cache hierarchy with a NVM. The key novelty of [210] is the exploition of the MESI cache coherence protocol to perform dynamic block reallocation between different cache partitions. Compared to the pure SRAM-based design, our hybrid scheme achieves 38% of energy saving with a mere 0.8% IPC degradation while extending the lifespan of STT-RAM partition at the same time. [72] proposes a Hybrid Scratch Pad Memory (HSPM) architecture which consists of SRAM and NVM to take advantage of the ultra-low leakage power, high density of NVM and fast read of SRAM. A novel data allocation algorithm as well as an algorithm to determine NVM/SRAM ratio for the novel HSPM architecture are proposed.

[109] proposes an asymmetric write architecture with redundant blocks (AWARE), that can improve the write latency by taking advantage of the asymmetric write characteristics of 1T-1MTJ STT-MRAM bit-cells. The asymmetry arises due to the nature of the storage element in STT-MRAM, wherein the time required for the two-state transitions (1 to 0 and 0 to 1) is not identical. In [179], the authors have investigated the behavior of GPGPU applications, where they present an efficient L2 cache architecture for GPUs based on STT-RAM technology. The STT-RAM L2 cache architecture proposed in this paper, can improve IPC by more than 100% (16% on average) while reducing the average consumed power by 20% compared to a conventional L2 cache architecture with equal on-chip area.

It is quite clear from the work being done in related areas, that NVMs haven't been looked into as options for the highest level, even for the data cache hierarchy very often. Also, unlike the others the main focus here is on bypassing the read latency limitations. Additionally, the write latency oriented techniques do not lead to good results and they do not really mitigate the real latency penalty. Since, the work is based on an ARM like general purpose processing platforms, the latency issues are crucial to the success of the overall system.

6.3 Replacing SRAM with STT-MRAM: Comparison with existing solutions

Table 6.1 compares the 64KB SRAM and STT-MRAM caches for the 32nm tech node with High Performance (HP) transistors. The STT-MRAM 64KB

Table 6.1: 64KB SRAM L1 D-cache vs 64KB STT-MRAM L1 D-cache (32nm tech node with High Performance (HP) transistors)

Parameters	SRAM	STT-MRAM
Read Latency	0.787ns	3.37ns
Write Latency	0.773ns	1.86ns
Leakage	204.76mW	28.35mW
Area	$146F^2$	$42F^2$
Associativity	2 way	2 way
Cache Line size	256 Bits	512 Bits

D-cache numbers that are used in our experiments are obtained by means of appropriate technology scaling and other optimizations applied to the advanced perpendicular dual MTJ cell with low power, high speed write operation and high magneto-resistive ratio, with the help of NVSim [46]. These numbers are consistent with the numbers presented by Samsung [93], Toshiba [147], Qualcomm etc. Unlike the bit-cell utilized in 5.3, the STT-MRAM bit cell here is a 2T-2MTJ complimentary type cell where the selector is planar CMOS-based. The interface is SRAM compatible. The standard supply voltage is 1V (with full V_{dd} across the BL to drive the switching of the STT-MRAM cell) and the interface is SRAM compatible. Depending on the gate oxide thickness the max voltage (overdrive) can extend upto 1.5V. We will require at-least 1.1V on the access transistor gate to generate the current required to switch the MTJ ($\sim 60\mu$A). The memory cell size is $0.1847\mu m^2$. The SRAM bit-cell is assumed to be 8T or higher. The use of the NVSim behavioural simulator for the energy, delay and area modeling of memories was motivated by its wide adoption in literature, availability and ease of use. It is to be noted that for smaller memory/cache sizes, similar to CACTI, NVSim numbers are quite pessimistic as a result of the abstraction level of the simulators and do not take into account the circuit modifications required to optimize a design for a particular NVM technology. Again, it must also be mentioned again that like in the previous chapters, these numbers were representative of the NVM technology maturity at the time and do not represent state of the art highlighted in table 2.2 (Chapter 2).

However, like it has been mentioned earlier, latency issues limit the use of STT-MRAM for higher level memories. As seen in table 6.1, the read latency of STT-MRAM is larger than it's SRAM counterpart by a factor of 3-4 for the instruction cache. Write latency issues can still be managed by techniques like the inclusion of a small L0 cache (like in the TMS320C64x/C64x DSP of Texas Instruments [203]), buffers ([198]) or by means of modifications such as the one proposed in [102]. A simple simulation can show that these latency issues, in particular read, have a major impact on performance when NVMs

are used in the first levels of the memory hierarchy, even for data caches that are not so read dependent like instruction caches. Here for instance, we have analyzed the impact of a STT-MRAM based D-Cache on the performance of a Single Core 1GHz ARM processor with 32KB IL1 cache, 64KB DL1 cache and a 2MB unified L2 cache. We utilize the micro-architecture simulator Gem5 [15] for this purpose and realistically assume that the read access time of the STT-MRAM cache is four times that of the usual SRAM cache, while write access is assumed to be twice that of the usual SRAM counterpart. We have used a subset of the PolyBench Benchmark suite [164] for our simulations.

Figure 6.1 shows the performance penalty on replacing just the SRAM D-cache by a NVM counterpart with similar characteristics (size, associativity...). The instruction cache and the unified L2 cache remain SRAM based. Even for the minimal read latency issue that is being considered here, we can observe a clear and unacceptably large performance overhead compared with the baseline. In-fact, '**reg-detect**' may suffer up to 55% performance penalty if the NVM D-cache is introduced instead of the regular SRAM one. The main conclusion of this analysis is that although STT-MRAM is a good candidate to replace SRAM D-caches given certain obvious advantages, a simple drop-in replacement is not advisable and some architecture modifications are required to reduce the impact of their latency limits.

6.4 Architectural modifications to address NVM limitations

In this chapter, we analyze the feasibility of using STT-MRAM for the L1 D-cache. We propose using a significantly modified version of a Very Wide Register (VWR) organization focused on improving SIMD access in SRAM caches (proposed in [173]). The enhanced intermediary Very Wide Buffer (VWB) that is proposed here is detailed in figure 6.2. The Very Wide Buffer here is an asymmetric register file organization. Micro-architecturally, a VWB is made of single ported cells. A post-decode circuit consisting of a multiplexer (MUX) is provided to select the appropriate word(s). The VWB is very close to logic and it can effectively act as a energy efficient high speed buffer between the DL1 and the processor. The interface of this register file organization is asymmetric: wide towards the memory and narrower towards the datapath. This is similar to a register file which is incorporated in the datapath pipeline cycles. The wide interface enables exploiting the locality of access of applications through wide loads from the memory to the VWB and also utilizes this bandwidth to hide latency. The wider memory array of the D-cache actually is more beneficial energy wise to the NVM here compared to that

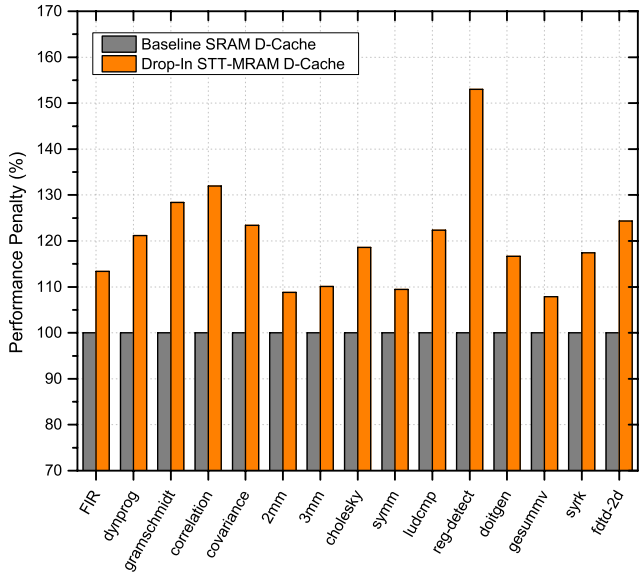


Figure 6.1: Performance penalty for the Drop in NVM D-Cache, relative to SRAM D-Cache. SRAM D-cache baseline = 100%.

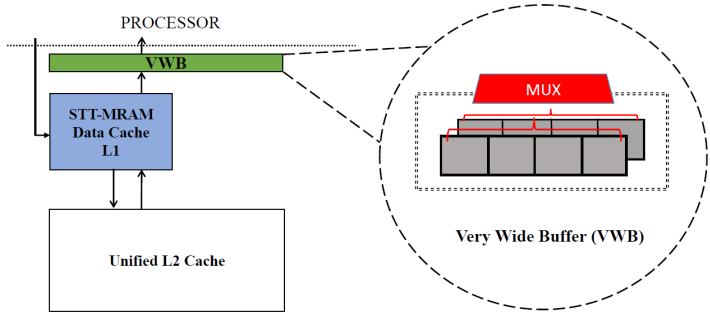


Figure 6.2: Modified L1 D-cache organization with a STT-MRAM D-cache and Very Wide Buffer in our General purpose platform ARM like platform.

of SRAM. This is because the cumulative capacitance of the SRAM transistors across a wide word will be much larger than that of it's NVM counterpart. This has a major effect on the energy and delay of the memory array. The VWB approach can be used in both data-parallel and non-data-parallel contexts.

While the VWB is very small in size, for practical purposes, it has to be able to

accommodate at-least 2KBit of data for efficient exploitation of data locality in our latency challenged scenario. The area gains offered by using a NVM make this feasible. The VWB in principle is used to hold critical and frequently executed data arriving from NVM DL1 and exploits data locality when the entire cache line is transferred into it. It must be noted that in the ARM like environment (based on the Cortex A9) that is our platform, in the data cache and instruction cache the word being read out is very wide and usually as big as the entire cache line. Hence, no need is present for any major circuit level modifications in our proposed implementation. The processor will read/write the appropriate word via the MUX selector of a single line from/to the VWB, thereby using a narrow interface matching the data-path of the processor. The STT-MRAM data memory will reads complete lines from the VWB, thereby using a wider interface which will increase the throughput to the memory.

The VWB is modeled like a fully associative buffer. It is made up of two lines of single ported cells in conjunction with each other in such a way that the data can be transferred from one line to another seamlessly back and forth. Both lines are connected to the post decoder MUX network. This way data can be written into and read from the VWB at the same time. Each VWB line has an associated tag. The load and store policies for the L1 data cache and the corresponding VWB in our proposal are as follows:

- Load Operation: The VWB is always checked for the data first in this scenario and if it is present then it's a hit and the data is read from it. Else, it is recorded as a miss and the NVM DL1 is checked. If the data is present, then it is read from the NVM DL1 and also written into the VWB. The evicted data from the VWB is stored in the NVM DL1. If the data is not present in the NVM DL1 also, then the miss is served from the next cache level, and the cache line containing the data block is then transferred into the processor and the VWB.
- Store Operation: The data block in the DL1 only updated via the VWB if it's already present in it. Otherwise, it's directly updated via the processor. A small write buffer is present when to hold the evicted data temporarily, while being transferred to the L2, when the data block in question has to be renewed. No write through is present to the L2 and main memory, and a write-back policy is implemented. If it's a miss, we follow the write allocate policy for the data cache array and a non allocate policy for the VWB. The data in the cache location is loaded in the block from the L2/main memory and this is followed by the write hit operation.

Regarding performance, the proposed mechanism de-couples the read hits from

the NVM, effectively removing the long latency read operation from the critical path. However, those reads may eventually happen when the VWB encounters a miss and the processor may try to fetch new data while the promotion of a cache line into the VWB is taking place (since the promotion may take as long as 4 cache cycles). We have simulated a banked NVM array, so no conflict will exist if both operations target different banks. Otherwise, the processor must be stalled, thus degrading system performance and potentially violating hard real-time operation. For the VWB organization proposed in this section the size and the implementation are the some of the few areas where tweaking to suit the platform, applications and other requirements are possible.

Figure 6.3 shows the effect of our micro-architectural modifications in reducing the penalty caused by NVM latency limitations. Although the reduction in penalty is significant, it's not not enough taking into account the fact that the benchmarks are minimal in size and aren't really representative of the heavy, robust or data intensive applications that will also be utilized in real time scenario's. Thus we need to cut the penalty further. This can be achieved through the exploitation of parallelization, cutting initial delay time to fetch critical data to the VWB and also cutting the delay by fetching repeatedly used data (in loops) into the VWB. These aspects are described in the next section.

6.5 Code Transformations and Optimization

Almost all modern systems try to utilize some form of parallelization nowadays for the efficient use of resources and greater performance. We try to exploit data level parallelism here by means of vectorization (essentially loop vectorization). Vectorization is a process wherein the program is converted from a scalar implementation, which processes a single pair of operands at a time, to a vector implementation, which processes one operation on multiple pairs of operands at once. For example, modern conventional computers, including specialized supercomputers, typically have vector operations that simultaneously perform operations such as four additions/subtractions etc. We identify the critical data and loops and vectorize them.

We broke down the contributions of the read and write access to the total penalty of the system for our NVM based proposal to enable us to use appropriate transformations suited to our problem (Figure 6.4). The read contribution towards the total penalty far exceeds that of the write. With increasingly complex kernels, the write penalty contribution also seems to increase, albeit slightly. However, the clear difference in impact between the

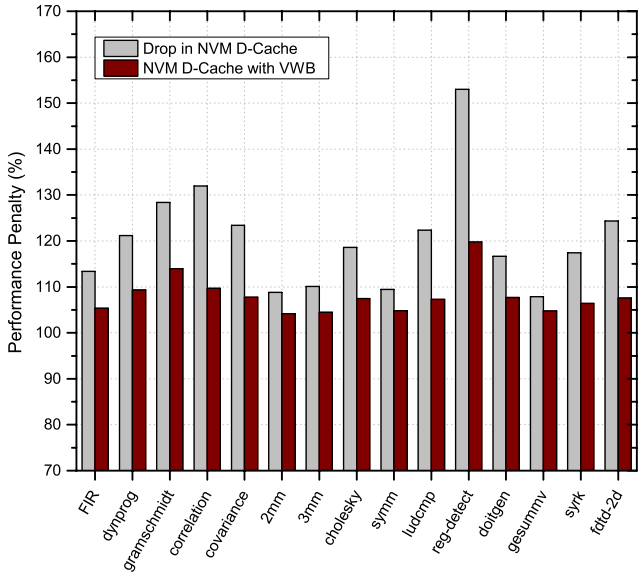


Figure 6.3: Performance penalty for the modified NVM D-Cache (with VWB) compared to a simple drop in NVM replacement. SRAM D-cache baseline = 100%.

two even in case of the data cache (as compared to the instruction cache where reads are much more critical) makes a case for the application of pre-fetching. Here, we can pre-fetch critical data and loop arrays to the VWB manually and hence reduce time taken to read it from the NVM. For the moment, we steer these transformations manually by the use of intrinsic functions to modify the individual Kernels.

Alignments of loops, jumps, pointers etc also help in reduction of penalty. We also attempt to transform conditional jumps in the innermost loops to branch-less equivalents, guess branch flow probabilities and try to reduce number of branches taken thus improving code locality. We carry out these optimizations automatically by specifying the individual intrinsic function flags at compile time for the different Benchmarks. Figure 6.5 details the effects of the above mentioned transformations and optimizations on the performance of our modified STT-MRAM based DL1 organization.

Another breakdown of the contribution to reduction of performance penalty, by means of code transformations (Figure 6.6), reveals that pre-fetching and vectorization have the largest positive impacts. Other intrinsic functions for

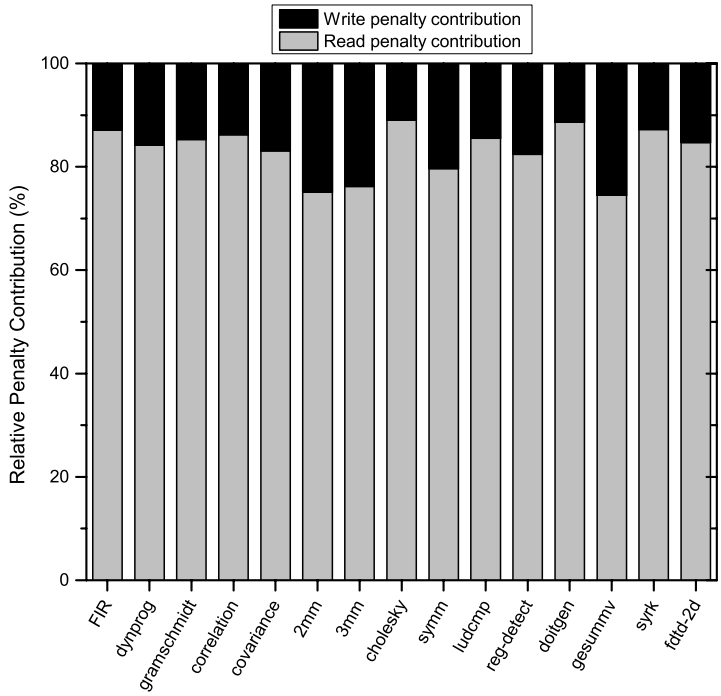


Figure 6.4: Performance penalty contribution of the read access latency and write access latency in the our modified NVM based D-cache proposal.

alignment, branch prediction and avoiding jumps etc become more significant as the kernel becomes larger and more complex. Predictably, pre-fetching is most impactful for the smallest kernels. A systematic approach is being looked into to facilitate and best exploit the above mentioned code transformations and optimizations in an appropriate manner. These will be employed uniformly after further explorations.

6.6 Exploration and Analysis

In order to evaluate the effectiveness of the proposed modifications, we implement our proposal and design methodologies via the GEM5 simulator ([15]). The use of the GEM5 architectural simulator was motivated by its wide adoption in literature, availability, accuracy and and our exploration space (cache architecture in general purpose RISC machine). We run all simulations

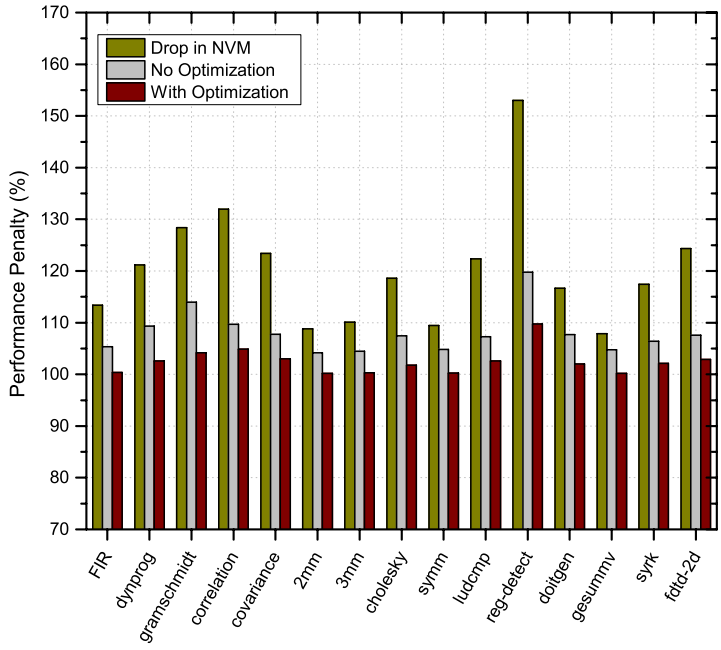


Figure 6.5: Performance penalty for the modified NVM DL1 (with VWB) with and without transformations and optimizations. SRAM D-cache baseline = 100%.

in the System-call Emulation (SE) mode with the arm detailed CPU type that mimics an ARM Cortex-A9 processor with a clock frequency of 1GHz. The cache configuration is set as follows: a 32KB 2-way associative L1 I-cache, a 64KB 2-way associative L1 D-cache and a 2MB 16-way associative unified L2 cache. For the simulations, we utilize a subset of the PolyBench, MediaBench and CPU SPEC2006 benchmark suite like before. Although the set of benchmarks tested here are not particularly large or heavily data intensive, it stands to reason that a fair extrapolation of these conditions even for larger benchmarks would produce significant reduction in the performance penalty induced by the introduction on an NVM cache in Level 1. The VWB size is usually taken to be 2KBit with 2 lines of 1KBit register files. The D-cache line size is 512 Bits.

One of our first explorations is to study the effect of the Size of the VWB on our proposed idea (Figure 6.7). It is quite clear from the figure that larger size VWB's help in reducing the penalty more. This is simply because of more code

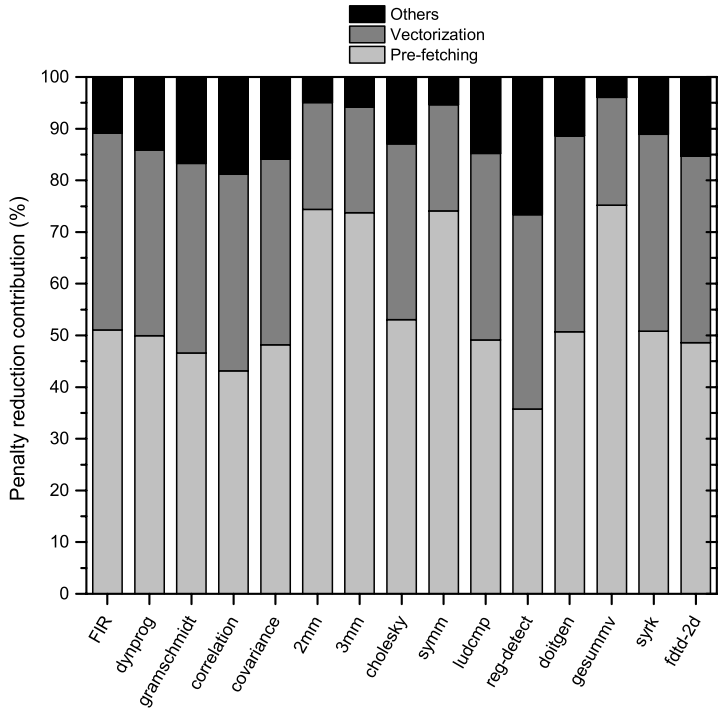


Figure 6.6: The contributions of the different transformations to the performance penalty reduction in our proposed NVM DLI (with VWB).

being able to fit into the VWB with it's increased capacity and hence lesser performance penalty. However, a limit is present to the VWB size put forward by technology, circuit level aspects cost and energy. The routing and layout also becomes cumbersome. Hence, we found it ideal to keep the size of the VWB to around 2KBit considering the area gains offered by the NVM. Keep in mind that a fully associative search also becomes a big problem with the increase in size of the VWB.

To have a better idea of where our proposal stands and it's effectiveness, we then compared it to a few techniques and proposals for write mitigation in NVMs like a variation of the commonly used L0 cache and the Enhaced MSHR presented in [102]. We try to utilize these proposals for the purpose of latency reduction in our scenario, where the read latency is more of a problem. The hardware structures are made fully associative and have the same size (2KBit) as that of the VWB for a fair comparison. However, the given structures are not as wide as the VWB and conform to the interface of the regular size memory

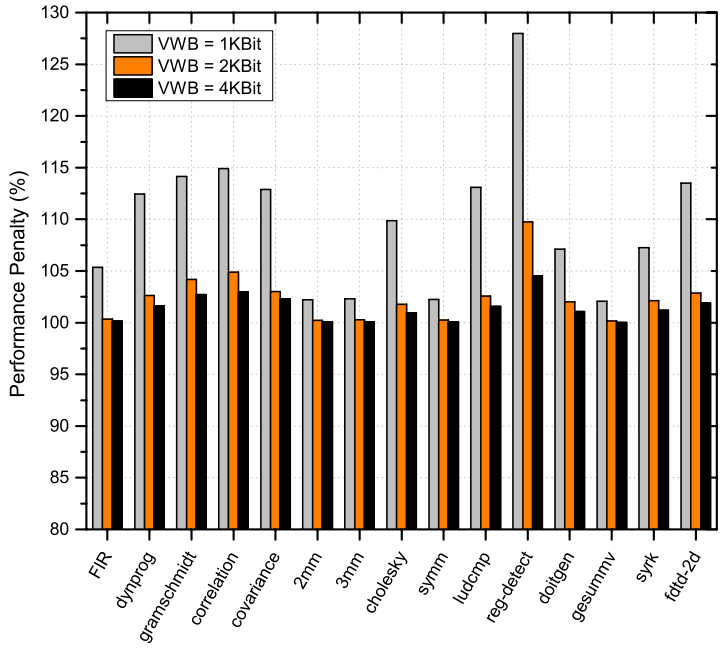


Figure 6.7: Performance penalty of the proposal for different sizes of the VWB in our proposal (optimized NVM DL1 with VWB). SRAM D-cache baseline = 100%.

Figure 6.8: Performance penalty comparisons between our proposal, a modified version of commonly employed L0 Cache and the EMSHR[102]. SRAM D-cache baseline = 100%.

array. The comparison is detailed in Figure 6.8. It's pretty evident that our proposal offers almost twice the penalty reduction as compared to the other previous proposals and this can be attributed to both the uniqueness of the structure and the effective software optimizations included to exploit it.

The effect of code transformations on the baseline SRAM based system is also a net positive. We have compared the net effect of the optimizations and transformations on the baseline system to that of our proposal (Figure 6.9). It stands to reason that while the software transformations can positively affect the baseline SRAM system (resulting in a better performance compared to our proposal by ∼8%), it is more pronounced in case of our NVM based proposal where the architecture and data allocation policy is tuned to exploit these

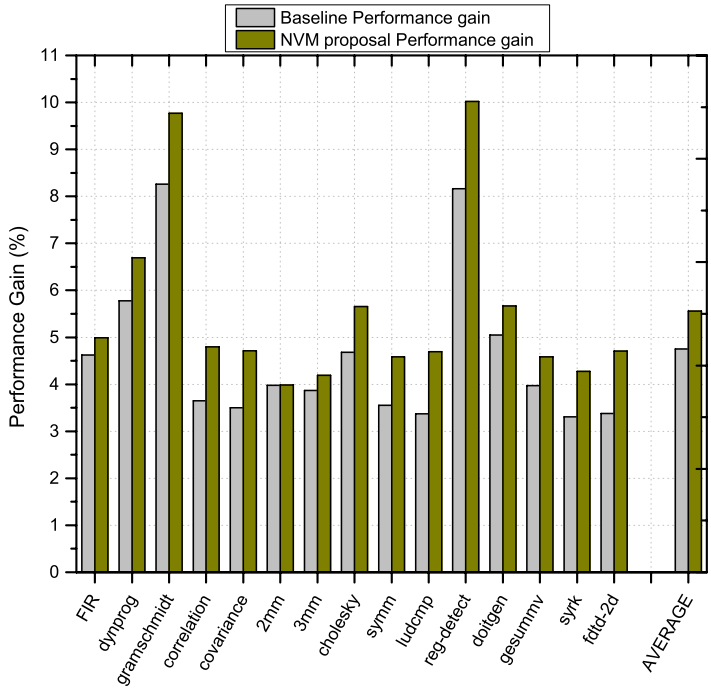


Figure 6.9: Effect of code transformations and optimizations on the baseline SRAM D-cache and a comparison with our proposal (NVM DL1 with VWB).

optimizations the most. Also, these do not take into account the obvious advantages offered by the NVM cache like more area leading to more capacity, and lower energy.

6.7 Conclusion

This chapter presents a modification of the traditional data cache organization by the inclusion of a NVM based cache instead of the traditional SRAM one along with the addition of a Very Wide Buffer for the effective exploitation of this STT-MRAM cache. This Very Wide Buffer also helps with the reduction of the performance penalty induced due to STT-MRAM latency limitations. Our explorations show that appropriate tuning of selective architecture parameters along with suitable data allocation techniques coupled with code transformations and optimizations can reduce the performance

penalty introduced by the NVM to extremely tolerable levels (8%) even in the worst cases. Furthermore, the use of NVMs also allows gains in area and even energy (power models have yet to be fully developed though). The gains in area can in turn can be utilized to accommodate D-caches with more capacity (around 2-3 times for STT-MRAM). Although the set of benchmarks tested here are not particularly large or heavily data intensive, it stands to reason that a fair extrapolation of these conditions even for larger benchmarks would result in a significant reduction of the performance penalty induced by the introduction of a NVM DL1.

Chapter 7

Conclusions and Future Work

7.1 Summary of Conclusions

A systematic exploration on the feasibility of incorporation of NVMs into higher levels of the memory hierarchy is presented. While the main focus of exploration is mostly relegated to the System Level methodologies and protocols like micro-architectural modifications, some circuit design and cell model aspects have also been looked at over the course of the PhD. In-fact, like it will be been conveyed in Section 7.1.1, it is only by means of a close co-optimization across all the different layers of abstraction that a realistic picture can be obtained about the feasibility for the emerging NVM technologies for the different application nodes and target systems. The framework highlighted in this thesis has been built and is being constantly improved by inputs from the design team (Sushil Sakhare, Oh HyungRock and Trong Hyunh Bao) and the device team (Woojin Kim and Siddharth Rao).

In this PhD, we have targeted low power applications and target systems are all timing sensitive with stringent constraints on throughput and/or latency, but mostly belong to the embedded context and less to the HPC (High Performance Computing) context. Three different target systems were considered for this study: an ultra low power embedded instruction memories for low power wireless/multimedia applications (Chapter 3), a custom Coarse-Grained Reconfigurable Architecture (CGRA) framework based on the ADRES platform (Chapter 4) and an ARM like single core general purpose system/ Mobile SoC (Chapter 5 and 6). The choice of NVM for our studies varied with the maturation of the NVM technology and it's effectiveness. Initially, while

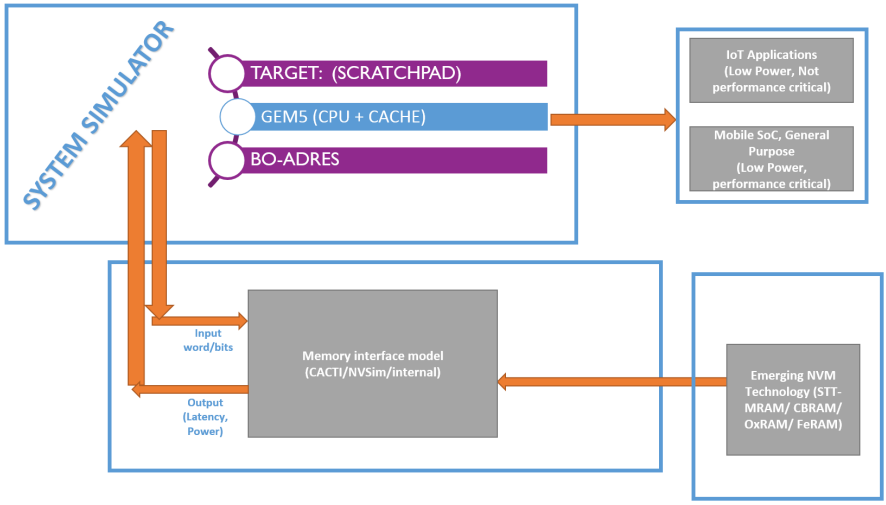


Figure 7.1: An overview of the methodology outlined in the PhD thesis to arrive at conclusions.

ReRAM (Chapter 3 and 4) was the preferred choice due to it's high R-Ratio and CMOS compatibility, the destructive process that defined the switching mechanisms for Resistive RAM lead to it falling short on multiple fronts over the course of time. This was in particular true for the limited endurance, high variability and the need for relatively high voltages (around 3-5V) for bit-cell operation. STT-MRAM (Chapter 5 and 6) has been the preferred choice of NVM for a good portion of the thesis and is the mature NVM technology currently suitable for our target applications and platforms. A flowchart for the methodology to arrive at our conclusions has been detailed in figure 7.1. This chapter will first highlight the conclusions of the different explorations detailed in the thesis. Then, we present in brief a framework to briefly evaluate a complete cross layer exploration of STT-MRAM based cache organization for an ultra low power platform. Finally, a look at the possible future work wraps up the thesis text.

7.1.1 eNVM based Instruction memory for an Ultra Low Power Platform

In chapter 3, we propose novel eNVM based instruction memory organizations, for such scratchpad based systems, that can address the challenges associated

with the write behaviour of the NVMs. We rethink the SRAM based instruction memory hierarchy for ultra low power embedded systems and replace it by an ReRAM based hybrid instruction memory organization. The instruction memory here resembles a Scratch-Pad Memory and other higher level software controlled memories. Initially, we analyze the circuit level modifications and optimizations required for the maximal effective use of NVMs (compared to SRAM) with asymmetric read/write (read faster and low energy compared to write). The circuit optimizations help with the transistor sizing, sense amplifier optimization and bit-line loading for the most optimized Energy-Delay product. We then propose a wide word NVM array that can be exploited as such. We then lay the groundwork for a novel NVM based instruction memory organization that has been micro-architecturally modified for the effective utilization of the NVM and other components. We select ReRAM as the NVM of choice for this exploration, since it's characteristics best suit the easy demonstration and implementation of our proposal. According to our analysis, even in the worst case scenario, the NVM based hybrid instruction memory organizations showed 87.02% reduction in read energy consumption at 0% performance penalty. Thus, a transition to NVM based memories is almost always advisable for the target applications running on ultra low power embedded systems.

7.1.2 eNVM based Configuration Memory Organization

In chapter 4, we extend the wide word ReRAM array presented earlier in chapter 3 for explorations on the configuration memory organization for CGRA interfaces. These interfaces are optimized for the large configuration words that are particular to CGRA architectures [1]. This affects design of the CGRA interface design significantly. We replace the traditional SRAM based interface by a eNVM based alternative. The eNVM based CGRA interface also re-utilizes the architectural modifications proposed for the scratchpad based ultra low power system to make NVM based architectures viable. Even in the worst case scenario's the most optimal eNVM based CGRA interface showed about 75% reduction in read energy consumption at 0% performance penalty. From the results it can be inferred that the proposed substitution of SRAM based L1 configuration memory with a eNVM based alternative has a clear advantage when it comes to energy consumption for the target applications in case of CGRAs.

7.1.3 NVM based I-Cache for High Performance General Purpose Processors

In chapter 5, we replace the highest layer of the SRAM based instruction cache hierarchy with an NVM based one for general purpose/mobile SoC target platforms. We propose a novel instruction cache configuration that address some of the challenges associated with the write behavior of NVMs by means of enhancements to the I-cache MSHR. Since the focus is on system level changes, we do not delve deeply into technology and circuit related matters in this chapter. Most of the circuit and system level calculations are based on the use of established simulators. We have focused on single core low power ARM like general purpose platforms here. We have used STT-MRAM for this exploration, since it's characteristics best suit the easy demonstration and implementation of our proposal. Our exploration shows that the appropriate tuning of selective architecture parameters can reduce the performance penalty introduced by the NVM to extremely tolerable levels ($\sim$1%). In addition, we show that we can also obtain significant reductions in energy consumption ($\sim$35%), given that the NVM technology has a favourable read energy per access when compared to SRAM. However, the pareto-optimum values for the different parameters are heavily application and platform dependent. Furthermore, the use of NVMs also allows gains in area and this in turn can be utilized to accommodate I-caches with more capacity ($\sim$2-3 times for STT-RAM). When configuring the modified NVM based I-cache to occupy the same area as the original SRAM-based configuration, the NVM based system outperforms the SRAM baseline and leads to even more energy savings. Finally, we also show that STT-MRAM memories are potentially good candidates to build feasible L1 I-caches even for relatively conservative scenarios.

7.1.4 STT-MRAM based D-cache explorations for High Performance General Purpose Processors

In chapter 6, we analyze the feasibility of using STT-MRAM for the L1 D-cache for general purpose/mobile SoC target platforms. We propose using a significantly modified version of a Very Wide Register (VWR) organization focused on improving SIMD access in SRAM caches (proposed in [173]). Our explorations show that appropriate tuning of selective architecture parameters along with suitable data allocation techniques coupled with code transformations and optimizations can reduce the performance penalty introduced by the NVM to extremely tolerable levels (8%) even in the worst cases. Furthermore, the use of NVMs also allows gains in area and even energy (power models have yet to be fully developed though). The gains in area can in

turn can be utilized to accommodate D-caches with more capacity (around 2-3 times for STT-MRAM) just like in the case of NVM based I-caches as shown in chapter 5. Although the set of benchmarks tested here is not particularly large or heavily data intensive, it stands to reason that a fair extrapolation of these conditions even for larger benchmarks would result in a significant reduction of the performance penalty induced by the introduction of a NVM DL1.

7.1.5 Case study: Cross Layer Exploration of a STT-MRAM based L2 cache memory organization

Due to the presence of numerous system level proposals and NVM based framework simulators presented in academia (and industry to a lesser extent), it becomes prudent to investigate their accuracy, effectiveness and ability to translate to effective designs that can be silicon verified for the products and real time performance. A number of the models and the papers covering them have been described in the previous chapters. The primary circuit framework used most widely for NVM arrays is usually NVSim [45]. There are also others like SCM-BSIM [226] which are based on similar C/C++ framework and model the latency and energy for accurately for the storage class memory platform. These NVM design frameworks are mere abstractions and they cannot be extrapolated/interpolated to get accurate numbers for the different levels of the memory hierarchy. Also, simple software based simulator frameworks simply cannot take into account the parasitics and margins associated with actual silicon based implementation. This is quite clear from the wide disparity apparent in the numbers from NVSim and the some of the industry data on NVM's (for instance [148, 146, 106, 156]).

For the device models, physics based TCAD and VerilogA compact models (like [218] [177] and [178]) are present, some of which are widely available [94] [219]. The are mostly physics based models that do not translate well when we look at the functionality and the behavior of the cell in an array since, there are always artifacts and the complex mechanisms at nano-dimensions are yet to be fully understood. Hence, we found it necessary to develop a complete framework that takes care of most of these issues and can be verified on silicon at multiple nodes and across the different layers (device compact model - circuit design and architecture - system). Finally, the design is incorporated at the system level and it's feasibility is analyzed for the ultra low power application domains. It must be stressed that the memory array designed in this case study is evaluated for the L2 cache level only.

MTJ device and Model

The high performance p-MTJ stack was developed on 300mm Si wafers according to process mentioned in [98]. The pMTJ stacks comprise of dual MgO interfaces interspersed by magnetic CoFeB electrodes, to improve the interfacial PMA. The free layer (FL) and reference layer (RL) consist of CoFeB-based multilayer stacks. The RL is coupled to a Co/Pt-based hard layer (HL) to form a synthetic antiferromagnet (iSAF), this allow us to keep our offset field near zero. The MTJ device used in our study has a pillar diameter of 20nm with the nominal TMR around 150% and resistance-area (RA) product of 10.9 $\Omega\mu m^2$ (from the CIPT measurement). We decided to follow a different approach for the compact model and build a purely analytic model based of the measured data and then relate it to the physics and behaviour of the device. This ensures a fast model that can be translated to silicon. For this we develop a practical measurement based analytical model of the MTJ.

STT-MRAM design and system analysis

The STT-MRAM design is realized at the 28nm TSMC technology node by utilizing the Cadence Virtuoso Analog Design Environment . The access transistor is planar NMOS. The 1T-1MTJ bit-cell design is most suitable for STT-MRAM design when the optimization targets are power and density as compared to the 2T-2MTJ (more area than eDRAM) and 3T-2MTJ (similar in area to SRAM) based designs. The bit cell dimensions are 260nm x 220nm. The standard supply voltage is 0.9V (with full V_{dd} across the BL to drive the switching of the STT-MRAM cell) and the interface is SRAM compatible. Depending on the gate oxide thickness the max voltage (overdrive) can extend upto 1.4V. We will require at-least 1.1V on the access transistor gate to generate the current required to switch the MTJ ($\sim$30μA). For the particular platform we are considering, performance isn't critical. Moreover, switching speeds that are deep in the precessional regime (where the thermal activation contribution to switching is minimal) can lead to high failure rate and the design will still be unable to match SRAM performance. A butterfly architecture is chosen for the layout of the memory macro (Figure 7.2). Each bit-cell is composed of a SL and BL parallel to each other, with the WL orthogonal to it. While the 1T-1MTJ MRAM bit-cell occupies only 1/4th the area of the conventional SRAM bit-cell, the complete STT-MRAM architecture is around 66.7% smaller than the SRAM architecture of similar capacity owing to the larger contribution of the periphery. Every IO consists of a MUX 16 to select an individual BL and it's corresponding SL for the read and write operation. During the read access, a single BL and the corresponding SL is biased out of 16 pairs to read and

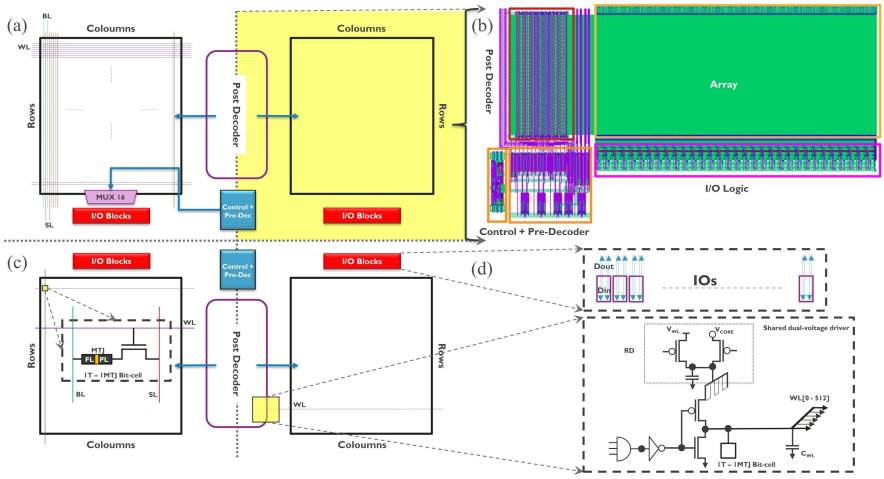


Figure 7.2: Schematic and architecture (a) Butterfly architecture (b) Layout Macro (c) Bit-cell schematic (d) Row Decoder and IO organization.

write a single bit per IO. When deselected, the SL and BL pairs are shorted to keep zero voltage across bit-cell, this keeps lowest standby power during read and write operation. The post-layout simulation was also carried out after full parasitic extraction.

Two different voltage levels for the WL are used during read and write operation to ensure lower current during read for same bias voltage across bit-cell (between BL and SL). The input data is written into MTJ by biasing BL and SL accordingly. The write driver voltage is optimized to write MTJ from P to AP (parallel to anti-parallel) state just before the onset of the precessional regime. The row decoder scheme is implemented utilizing a common dual-voltage driver. 512 (WLs) x 1024 (BLs) bit-cells are present per slice of butterfly excluding reference cells. V_{dd} supply voltage is taken to be 0.9V. The sense amplifier is enabled when the difference between the reference and the cell read out reaches around 70mV. The TMR is taken to be 100% in-order to account for any random, systematic and parametric defects on silicon. The write operation is performed at V_{BL} (V_{dd}), whereas the read is performed at a lower threshold.

Finally, to evaluate our design we compared it to the commercially available SRAM compiler for the 28nm node (Fig. 7.3). It's quite clear from the graph that our STT-MRAM design is more energy efficient (for L2 and L3 caches in the target application domains). In-fact the write energy of our proposed

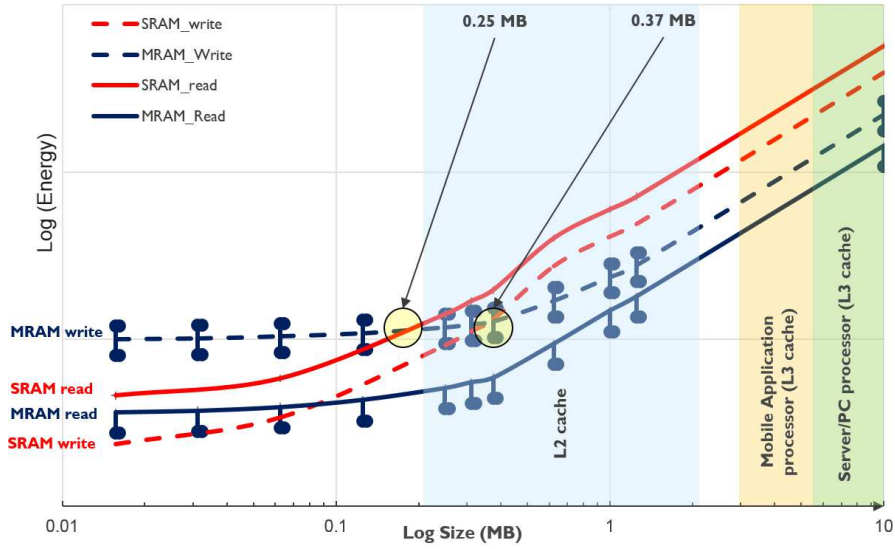


Figure 7.3: SRAM vs STT-MRAM energy consumption with error bars for PT impact.

STT-MRAM design falls short of SRAM read at around 0.25MB and SRAM write at around 0.37 MB. This is the first such study to show such gains for a simple drop-in replacement without any micro-architectural modifications even for very small sizes ($\sim$ 8KB onwards we see reasonable gain of STT-MRAM read vs SRAM read). These gains can be attributed to the extremely low contribution of the STT-MRAM array to total energy, standby power for SRAM (both write and read accesses) and superior MTJ device characteristics. As far as the latencies are considered, the read latency varies from 2.5 to 4.7 ns and the write latency for AP2P and P2AP are around 3.5ns and 9.1ns respectively. The read latency is dominated by the RC delay and time taken to generate the differential (70mV) for the sense amplifier. For the write latency, the device switching time is the dominant factor.

In order to evaluate the effectiveness of the proposed modifications, we implement our proposal and design methodologies via the GEM5 simulator ([15]). We run all simulations in the System-call Emulation (SE) mode and mimic an ARM Cortex-A9 processor with a clock frequency of 1GHz. The cache configuration is set as follows: a 32KB 2-way associative L1 I-cache, a 64KB 2-way associative L1 D-cache and a 2MB 16-way associative unified L2 cache. We propose a direct drop-in replacement for the L2 cache SRAM via our STT-MRAM design. For the simulations, we utilize a subset of

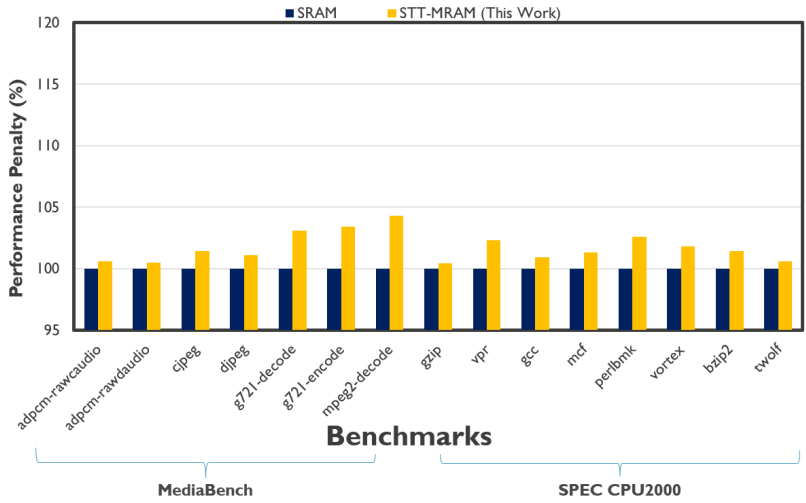


Figure 7.4: Performance penalty for drop-in STT-MRAM L2 cache vs SRAM.

the Mediabench and SPEC CPU20006 benchmark suites. The use of the GEM5 architectural simulator was motivated by its wide adoption in literature, availability, accuracy and and our exploration space (cache architecture in general purpose RISC machine).

For a very negligible performance penalty (<5%), we show upto 85% reduction in the L2 cache energy at one third the area. This is the first silicon realizable design and analysis on the 28nm node for STT-MRAM with such a favourable outlook even for very small capacities in terms of energy and area benefits. This clearly shows that STT-MRAM is very promising for the ultra low power application domains. Another important conclusion that can be drawn from results highlighted in Fig. 7.3 is with regards to the place of STT-MRAM in the memory hierarchy. The utilization of STT-MRAM area benefits can clearly lead to lesser cache miss rates (both L2 and L3) and effectively result in a more efficient system even performance wise (as shown in the chapters: Chapter 5 and 6). The case study is presented in more detail in [103].

7.1.6 Impact of multicore architectures

While this thesis has only dealt with single core architectures, it is important to mention about multi-core architectures (since they are widely used now even for the low power platforms) and how our proposed architectural modifications

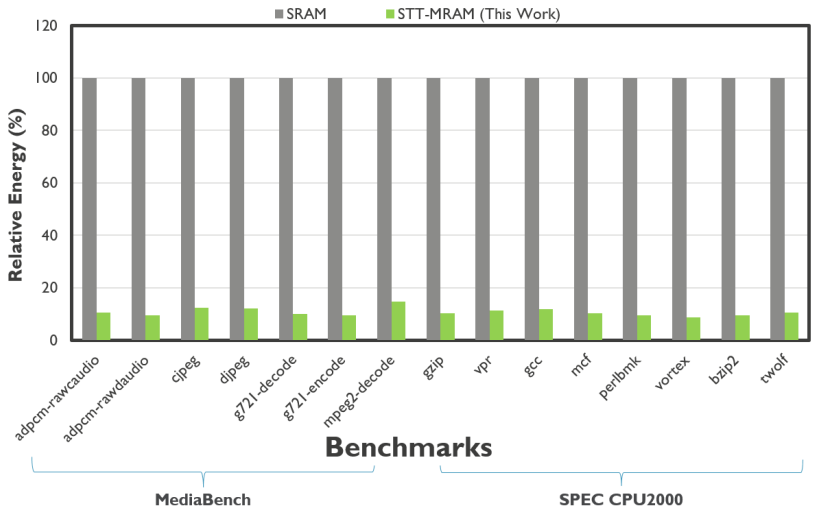


Figure 7.5: Energy relative to SRAM for drop-in STT-MRAM.

to facilitate NVMS will feature in them. The first thing to consider is level at which the NVM will be incorporated into the hardware/software controlled cache hierarchy. From Intel's Sandybridge micro-architecture release for the 32nm technology node (January 2011)vto the present Kaby Lake series, L1 and L2 caches have been dedicated to individual cores, with the L3 caches onward being shared. In such a scenario, our modifications to facilitate NVM incorporation at L1 (or even a drop-in at L2) do not face serious coherency issues. These systems, where a dedicated cache for each processor, core or node is used, a consistency problem may occur when a same data is stored in more than one cache. This problem arises when a data is modified in one cache. This problem can be solved in two ways:

- Invalidate all the copies on other caches (broadcast-invalidate).
- Update all the copies on other caches (write-broadcasting), while the memory may be updated (write through) or not updated (write-back).

Since, we don't consider large multi-core systems in our exploration space, snoopy caches with dedicated coherency controllers could take care of any data consistency. If we consider the extension of our NVM into the L3 caches, the negligible to nil performance penalty encountered would simply render any special methods of coherency control/data consistency checks unnecessary.

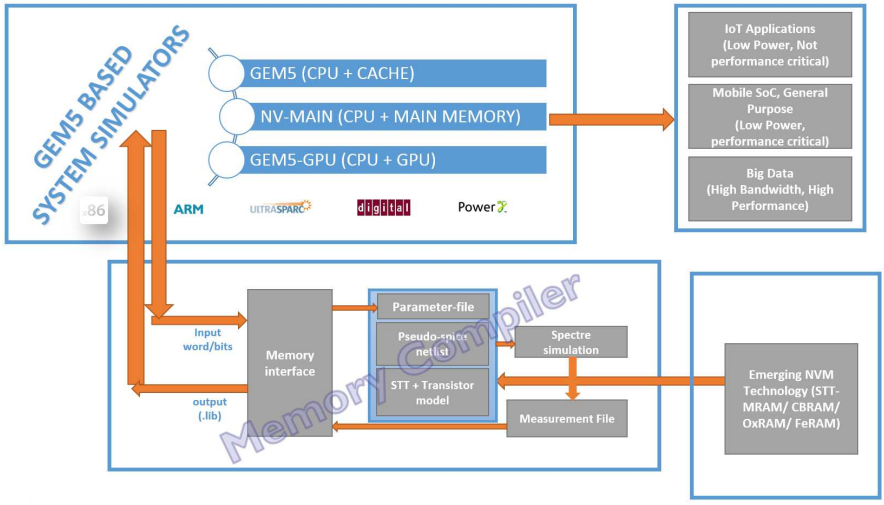


Figure 7.6: A general overview of the future framework to check the feasibility of an NVM technology for different target applications and platforms.

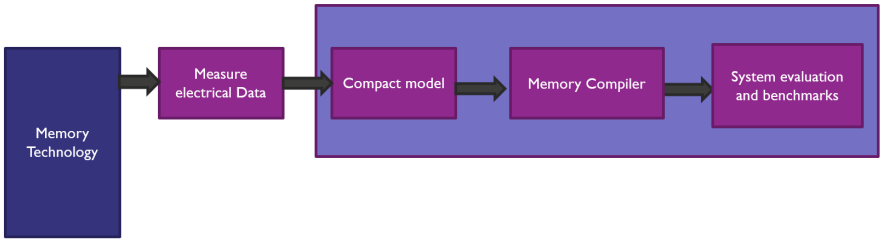


Figure 7.7: The feedback loop from system analysis to re-target the NVM technology and optimize it as per requirements.

7.2 Future Work

It is quite clear from the conclusions listed in the earlier section for the different target platforms and applications that it is only by means of a close co-optimization across all the different layers of abstraction that a realistic picture can be obtained about the feasibility for the emerging NVM technologies to go along with maximum gains. This means that the memory interface which incorporates the array design and compact modeling is crucial to system level analysis. Also, a feedback loop is essential to retarget the NVM from technology

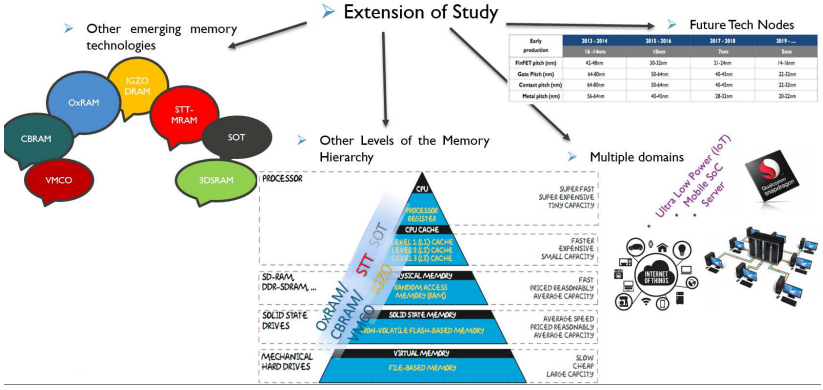


Figure 7.8: The different venues for extending the study.

aspect i.e. to explore the introduction of a new NVM technology based on the obtained memory organization exploration results. So this necessitates a what-if exploration framework. The required changes in the framework and the feedback loop are highlighted in figures 7.6 and 7.7. The future framework must ideally include a memory compiler to generate the power, latency and area for multiple NVM technologies and also for the different (future) technology nodes. From the system level point of view, the system simulator must be able to simulate different memory organizations, platforms, architectures and run representative workloads targeting varied application domains such as IoT sensor networks, mobile SoCs, gateways and Big Data servers. This should be coupled with a global Pareto trade-off analysis to arrive at future promising road-maps per major application domain. From our analysis, we have also concluded that Gem5 based system simulators are currently best suited from all perspectives to handle our requirements. Finally, figure 7.8 provides a broad overview of the different venues for the extensions of the study from this PhD. Apart from the above mentioned areas of exploration, a more concerted effort is also required to explore architectural and circuit-level solutions for mitigating the deficiencies of the current NVM memory technologies like lower endurance, voltage scalability, area scalability and variability.

Curriculum vitae

PERSONAL INFORMATION

Name : Manu Perumkunnil Komalan

Date of Birth : 14th April 1988

Residence : Leeuwerikenstraat 57, Apartment 504, Heverlee 3001, Belgium

Nationality : Indian

Email : Manu.Perumkunnil@imec.be

Mobile : 0032-489417279/ 0034-659785949

EDUCATION

PhD in Computer Science : 2011 Sept- 2017 March (Tentative)

Universidad Complutense de Madrid, Madrid, Spain
KU Leuven, Leuven, Belgium
Specialization: Non Volatile Memory Systems

Integrated Masters in Nanotechnology : 2005 Sept- 2011 January

Amity University, Noida, India
Specialization: Nanoelectronics

TRAINING

Imec academy, Imec, Belgium
FEOL and BEOL process flow, Beyond CMOS, Nanoscale CMOS Process

Technology, Nanometer CMOS IC's

HiPEAC
Advanced Computer Architecture Compilation for High-Performance Embedded Systems (ACACES) Summer School and Workshop.

Sun Microsystems (Oracle), India
Development of a Software Defined Radio on Open SPARC T2 platform

Indian Institute of Science, Bangalore, India
Semiconductor Physics Opto-electronics.

JNU, New Delhi, India
Design and Simulation of IC's (Alliance CAD tools)

SPIE Indian Institute of Technology - Bombay, India
Photonics Workshop

Bibliography

[1] Aa, T., et al. Micro-architectural optimization of a coarse-grained array based baseband processor. In *Signal Processing Systems (SIPS), 2010 IEEE Workshop on* (2010), IMEC, pp. 146–150. pages 94, 97, 160

[2] Ahn, S. J., et al. Highly manufacturable high density phase change memory of 64mb and beyond. In *IEEE International Electron Devices Meeting Technical Digest (IEDM), 2004* (December 2004), Samsung, pp. 907–910. pages 34

[3] Ambrogio, S., Balatti, S., Cubeta, A., Calderoni, A., Ramaswamy, N., and Ielmini, D. Understanding switching variability and random telegraph noise in resistive ram. In *2013 IEEE International Electron Devices Meeting* (Dec 2013), pp. 31.5.1–31.5.4. pages 124

[4] Apalkov, D., Khvalkovskiy, A., Watts, S., Nikitin, V., Tang, X., Lottis, D., Moon, K., Luo, X., Chen, E., Ong, A., Driskill-Smith, A., and Krounbi, M. Spin-transfer torque magnetic random access memory (stt-mram). *J. Emerg. Technol. Comput. Syst. 9*, 2 (May 2013), 13:1–13:35. pages vii, 27

[5] Apalkov, D., Khvalkovskiy, A., Watts, S., Nikitin, V., Tang, X., Lottis, D., Moon, K., Luo, X., Chen, E., Ong, A., Driskill-Smith, A., and Krounbi, M. Spin-transfer torque magnetic random access memory (stt-mram). *J. Emerg. Technol. Comput. Syst. 9*, 2 (May 2013), 13:1–13:35. pages 11, 27, 124, 144

[6] ARM. Cortex a9 technical reference manual. Tech. rep., ARM, 2008. pages xi, 127, 129

[7] Artes, A., Ayala, J., Sathanur, A., Huisken, J., and Catthoor, F. Run-time self-tuning banked loop buffer architecture for power optimization of dynamic workload applications. In *19th International*

Conference on VLSI and System-on-Chip, VLSI-SoC 2011 (October 2011), pp. 136–141. pages 65

[8] Artes, A., Ayala, J. L., Sathanur, A. V., Huisken, J., and Catthoor, F. Run-time self-tuning banked loop buffer architecture for power optimization of dynamic workload applications. In *2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip* (Oct 2011), pp. 136–141. pages viii, 64, 65

[9] Atwood, G., and Bez, R. Current status of chalcogenide phase change memory. In *Device Research Conference Digest, 2005. DRC '05. 63rd* (June 2005), vol. 1, pp. 29–33. pages 34

[10] Bajwa, R., Hiraki, M., Kojima, H., Gorny, D., Nitta, K., Shridhar, A., Seki, K., and Sasaki, K. Instruction buffering to reduce power in processors for signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 5*, 4 (December 1997), 417–424. pages 65, 98

[11] Bamal, M., Grossar, E., Stucchi, M., and Maex, K. Interconnect width selection for deep submicron designs using the table lookup method. In *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction* (New York, NY, USA, 2004), SLIP '04, ACM, pp. 41–44. pages 4

[12] Bellas, N., Hajj, I., Polychronopoulos, C., and Stamoulis, G. Architectural and compiler techniques for energy reduction in high-performance microprocessors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 8*, 3 (June 2000), 317–326. pages 65, 98

[13] Berg, C. PLRU Cache Domino Effects. In *6th International Workshop on Worst-Case Execution Time Analysis (WCET'06)* (Dagstuhl, Germany, 2006), F. Mueller, Ed., vol. 4 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. pages 53

[14] Bhavnagarwala, A. J., Tang, X., and Meindl, J. D. The impact of intrinsic device fluctuations on cmos sram cell stability. *IEEE Journal of Solid-State Circuits 36*, 4 (Apr 2001), 658–665. pages 4

[15] Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. The gem5 simulator. *SIGARCH Comput. Archit. News 39*, 2 (Aug. 2011), 1–7. pages 125, 131, 147, 152, 165

[16] Burr, G., Kurdi, B., Scott, J., Lam, C., Gopalakrishnan, K., and Shenoy, R. Overview of candidate device technologies for storage-class memory. *IBM Journal of Research and Development 52*, 4.5 (July 2008), 449–464. pages xv, 5, 9, 24, 41, 43, 44

[17] Calderoni, A., Sills, S., and Ramaswamy, N. Performance comparison of o-based and cu-based reram for high-density applications. In *2014 IEEE 6th International Memory Workshop (IMW)* (May 2014), pp. 1–4. pages 125

[18] Cappelletti, P., Golla, C., Olivo, P., and Zanoni, E. *Flash Memories*, vol. 1. Springer US, 1999. pages 23

[19] Chai, Y., Wu, Y., Takei, K., Chen, H.-Y., Yu, S., Chan, P. C. H., Javey, A., and Wong, H. S. P. Resistive switching of carbon-based rram with cnt electrodes for ultra-dense memory. In *Electron Devices Meeting (IEDM), 2010 IEEE International* (Dec 2010), pp. 9.3.1–9.3.4. pages 32

[20] Chatterjee, N., Shevgoor, M., Balasubramonian, R., Davis, A., Fang, Z., Illikkal, R., and Iyer, R. Leveraging heterogeneity in dram main memories to accelerate critical word access. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture* (Washington, DC, USA, 2012), MICRO-45, IEEE Computer Society, pp. 13–24. pages 45

[21] Chen, A., Haddad, S., Wu, Y.-C., Fang, T.-N., Lan, Z., Avanzino, S., Pangrle, S., Buynoski, M., Rathor, M., Cai, W., Tripsas, N., Bill, C., Buskirk, M. V., and Taguchi, M. Non-volatile resistive switching for advanced memory applications. In *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.* (Dec 2005), pp. 746–749. pages 30

[22] Chen, E., Apalkov, D., Diao, Z., Driskill-Smith, A., Druist, D., Lottis, D., Nikitin, V., Tang, X., Watts, S., Wang, S., Wolf, S. A., Ghosh, A. W., Lu, J. W., Poon, S. J., Stan, M., Butler, W. H., Gupta, S., Mewes, C. K. A., Mewes, T., and Visscher, P. B. Advances and future prospects of spin-transfer torque random access memory. *IEEE Transactions on Magnetics 46*, 6 (June 2010), 1873–1878. pages 26

[23] Chen, Y., and Petti, C. Reram technology evolution for storage class memory application. In *2016 46th European Solid-State Device Research Conference (ESSDERC)* (Sept 2016), pp. 432–435. pages xv, 43, 44, 46, 48, 51

[24] Chen, Y.-T., Cong, J., Huang, H., Liu, B., Liu, C., Potkonjak, M., and Reinman, G. Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012* (March 2012), pp. 45–50. pages 47, 95

[25] Chen, Y. Y., Govoreanu, B., Goux, L., Degraeve, R., Fantini, A., Kar, G. S., Wouters, D. J., Groeseneken, G., Kittl, J. A., Jurczak, M., and Altimime, L. Balancing set/reset pulse for $> 10^{10}$ endurance in hfo_2/hf 1t1r bipolar rram. *IEEE Transactions on Electron Devices 59*, 12 (Dec 2012), 3243–3249. pages 32, 58, 100

[26] Chhabra, S., and Solihin, Y. i-nvmm: A secure non-volatile main memory system with incremental encryption. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on* (June 2011), pp. 177–188. pages 47

[27] Cho, S. H., Lee, D. I., Jung, J. H., and Kim, T. W. Electrical bistabilities and memory stabilities of nonvolatile bistable devices fabricated utilizing c 60 molecules embedded in a polymethyl methacrylate layer. *Nanotechnology 20*, 34 (2009), 345204. pages 41

[28] Choi, B. J., Jeong, D. S., Kim, S. K., Rohde, C., Choi, S., Oh, J. H., Kim, H. J., Hwang, C. S., Szot, K., Waser, R., Reichenberg, B., and Tiedke, S. Resistive switching mechanism of tio2 thin films grown by atomic-layer deposition. *Journal of Applied Physics 98*, 3 (2005). pages 30

[29] Chun, K., Zhao, H., Harms, J., Kim, T., Wang, J., and Kim, C. A scaling roadmap and performance evaluation of in-plane and perpendicular mtj based stt-mrams for high-density cache memory. *IEEE Journal of Solid-State Circuits 48*, 2 (2013), 598–610. pages 27

[30] Chun, K. C., Zhao, H., Harms, J. D., Kim, T. H., Wang, J. P., and Kim, C. H. A scaling roadmap and performance evaluation of in-plane and perpendicular mtj based stt-mrams for high-density cache memory. *IEEE Journal of Solid-State Circuits 48*, 2 (Feb 2013), 598–610. pages 121

[31] Clark, L., Hoffman, E., Miller, J., Biyani, M., Liao, Y., Strazdus, S., Morrow, M., Velarde, K., and Yarch, M. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid-State Circuits 36*, 11 (11 2001), 1599–1608. pages 2

[32] Condit, J., Nightingale, E. B., Frost, C., Ipek, E., Lee, B., Burger, D., and Coetzee, D. Better i/o through byte-addressable, persistent memory. In *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles* (New York, NY, USA, 2009), SOSP '09, ACM, pp. 133–146. pages 47

[33] Corp, R. I. Ramtron announces 8-megabit parallel nonvolatile f-ram memory, 2009. pages 37

[34] Corp, T. Toshiba develops world's highest-bandwidth, highest density non-volatile ram, 2009. pages 37

[35] Cosemans, S., Dehaene, W., and Catthoor, F. A 3.6 pj/access 480 mhz, 128 kb on-chip sram with 850 mhz boost mode in 90nm cmos with tunable sense amplifiers. *IEEE Journal of Solid-State Circuits 44*, 7 (July 2009), 2065–2077. pages 59, 60

[36] Cronquist, D. C., Fisher, C., Figueroa, M., Franklin, P., and Ebeling, C. Architecture design of reconfigurable pipelined datapaths. In *Advanced Research in VLSI, 1999. Proceedings. 20th Anniversary Conference on* (Mar 1999), pp. 23–40. pages 94

[37] Cubukcu, M., Boulle, O., Drouard, M., Garello, K., Avci, C. O., Miron, I. M., Langer, J., Ocker, B., Gambardella, P., and Gaudin, G. Spin-orbit torque magnetization switching of a three-terminal perpendicular magnetic tunnel junction. *Applied Physics Letters 104*, 4 (2014), 042406. pages 28, 29

[38] David Tarjan, Shyamkumar Thoziyoor, N. P. J. Cacti 4.0: Technical report, 2006. pages 72

[39] Dawber, M., Rabe, K. M., and Scott, J. F. Physics of thin-film ferroelectric oxides. *Rev. Mod. Phys. 77* (Oct 2005), 1083–1130. pages 37

[40] Declerck, G. A look into the future of nanoelectronics. In *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on* (June 2005), pp. 6–10. pages 4

[41] Degraeve, R., Fantini, A., Roussel, P., Goux, L., Costantino, A., Chen, C. Y., Clima, S., Govoreanu, B., Linten, D., Thean, A., and Jurczak, M. Quantitative endurance failure model for filamentary rram. In *VLSI Technology (VLSI Technology), 2015 Symposium on* (June 2015), pp. T188–T189. pages 124

[42] Dhiman, G., Ayoub, R., and Rosing, T. Pdram: A hybrid pram and dram main memory system. In *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE* (July 2009), pp. 664–669. pages 45

[43] Dong, X., Muralimanohar, N., Jouppi, N., Kaufmann, R., and Xie, Y. Leveraging 3d pcram technologies to reduce checkpoint overhead for future exascale systems. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis* (New York, NY, USA, 2009), SC '09, ACM, pp. 57:1–57:12. pages 47

[44] Dong, X., Wu, X., Sun, G., Xie, Y., Li, H., and Chen, Y. Circuit and microarchitecture evaluation of 3d stacking magnetic ram (mram) as a universal memory replacement. In *2008 45th ACM/IEEE Design Automation Conference* (June 2008), pp. 554–559. pages 28

[45] Dong, X., Xu, C., Xie, Y., and Jouppi, N. Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 31*, 7 (July 2012), 994–1007. pages xv, 46, 48, 51, 83, 162

[46] Dong, X., Xu, C., Xie, Y., and Jouppi, N. Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 31*, 7 (2012), 994–1007. pages 120, 146

[47] Farkas, K., and Jouppi, N. Complexity/performance tradeoffs with non-blocking loads. In *Computer Architecture, 1994., Proceedings the 21st Annual International Symposium on* (1994), pp. 211–222. pages 128

[48] Fasthuber, R. *An Energy-Efficient Architecture Template for Wireless Communication Systems*. PhD thesis, KU Leuven, 2012. pages vii, 3

[49] for Semiconductors, I. T. R. Itrs roadmap 2006, 2006. pages 37

[50] Fujimoto, M., Koyama, H., Hosoi, Y., Ishihara, K., and Kobayashi, S. High-speed resistive switching of tio 2 /tin nano-crystalline thin film. *Japanese Journal of Applied Physics 45*, 3L (2006), L310. pages 30

[51] Gallagher, W. J., and Parkin, S. S. P. Development of the magnetic tunnel junction mram at ibm: From first junctions to a 16-mb mram demonstrator chip. *IBM Journal of Research and Development 50*, 1 (Jan 2006), 5–23. pages 25

[52] Gambardella, P., and Miron, I. M. Current-induced spin–orbit torques. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 369*, 1948 (2011), 3175–3197. pages 28

[53] Gielen, G., and Dehaene, W. Analog and digital circuit design in 65 nm cmos: end of the road? In *Design, Automation and Test in Europe, 2005. Proceedings* (March 2005), pp. 37–42 Vol. 1. pages 4

[54] Goldstein, S. C., Schmit, H., Budiu, M., Cadambi, S., Moe, M., and Taylor, R. R. Piperench: a reconfigurable architecture and compiler. *Computer 33*, 4 (Apr 2000), 70–77. pages 94

[55] Gopal, J. K., Do, A. T., Singh, P., Chua, G. L., and Kim, T. T. H. A cantilever-based nem nonvolatile memory utilizing electrostatic actuation and vibrational deactuation for high-temperature operation. *IEEE Transactions on Electron Devices 61*, 6 (June 2014), 2177–2185. pages 42

[56] Gordon-Ross, A., and Vahid, F. Dynamic loop caching meets preloaded loop caching - a hybrid approach. In *20th International Conference on Computer Design (ICCD 2002)* (2002), pp. 446–449. pages 98

[57] Goux, L., Fantini, A., Kar, G., Chen, Y. Y., Jossart, N., Degraeve, R., Clima, S., Govoreanu, B., Lorenzo, G., Pourtois, G., Wouters, D. J., Kittl, J. A., Altimime, L., and Jurczak, M. Ultralow sub-500na operating current high-performance tinal2o3hfo2hftin bipolar rram achieved through understanding-based stack-engineering. In *VLSI Technology (VLSIT), 2012 Symposium on* (June 2012), pp. 159–160. pages 33

[58] Govoreanu, B., Crotti, D., Subhechha, S., Zhang, L., Chen, Y. Y., Clima, S., Paraschiv, V., Hody, H., Adelmann, C., Popovici, M., Richard, O., and Jurczak, M. A-vmco: A novel forming-free, self-rectifying, analog memory cell with low-current operation, nonfilamentary switching and excellent variability. In *2015 Symposium on VLSI Technology (VLSI Technology)* (June 2015), pp. T132–T133. pages 33

[59] Govoreanu, B., Kar, G., Chen, Y., Paraschiv, V., Kubicek, S., Fantini, A., Radu, I. P., Goux, L., Clima, S., Degraeve, R., Jossart, N., Richard, O., Vandeweyer, T., Seo, K., Hendrickx, P., Pourtois, G., Bender, H., Altimime, L., Wouters, D., Kittl, J., and Jurczak, M. 10x10nm2 hf/hfox crossbar resistive ram with

excellent performance, reliability and low-energy operation. In *IEEE International Electron Devices Meeting Technical Digest (IEDM), 2011* (December 2011), imec, pp. 729–732. pages 67, 100

[60] Govoreanu, B., Redolfi, A., Zhang, L., Adelmann, C., Popovici, M., Clima, S., Hody, H., Paraschiv, V., Radu, I. P., Franquet, A., Liu, J. C., Swerts, J., Richard, O., Bender, H., Altimime, L., and Jurczak, M. Vacancy-modulated conductive oxide resistive ram (vmco-rram): An area-scalable switching current, self-compliant, highly nonlinear and wide on/off-window resistive switching cell. In *2013 IEEE International Electron Devices Meeting* (Dec 2013), pp. 10.2.1–10.2.4. pages 33

[61] Graphics, M. Modelsim se 10.0. pages 74

[62] Guo, Y., Zhuge, Q., Hu, J., Qiu, M., and Sha, E. H. M. Optimal data allocation for scratch-pad memory on embedded multi-core systems. In *2011 International Conference on Parallel Processing* (Sept 2011), pp. 464–471. pages 57

[63] Guy, J., Molas, G., Blaise, P., Carabasse, C., Bernard, M., Roule, A., Carval, G. L., Sousa, V., Grampeix, H., Delaye, V., Toffoli, A., Cluzel, J., Brianceau, P., Pollet, O., Balan, V., Barraud, S., Cueto, O., Ghibaudo, G., Clermidy, F., Salvo, B. D., and Perniola, L. Experimental and theoretical understanding of forming, set and reset operations in conductive bridge ram (cbram) for memory stack optimization. In *2014 IEEE International Electron Devices Meeting* (Dec 2014), pp. 6.5.1–6.5.4. pages 32

[64] Haensch, W., Nowak, E. J., Dennard, R. H., Solomon, P. M., Bryant, A., Dokumaci, O. H., Kumar, A., Wang, X., Johnson, J. B., and Fischetti, M. V. Silicon cmos devices beyond scaling. *IBM Journal of Research and Development 50*, 4.5 (July 2006), 339–361. pages 4

[65] Hagen, J. A., Li, W., Steckl, A. J., and Grote, J. G. Enhanced emission efficiency in organic light-emitting diodes using deoxyribonucleic acid complex as an electron blocking layer. *Applied Physics Letters 88*, 17 (2006). pages 41

[66] Heckmann, R., Langenbach, M., Thesing, S., and Wilhelm, R. The influence of processor architecture on the design and the results of wcet tools. *Proceedings of the IEEE 91*, 7 (July 2003), 1038–1054. pages 52, 53

[67] Hong, S., Auciello, O., and Wouters, D. *Emerging Non-Volatile Memories*, 1 ed. Springer US, 2014. pages xv, 43, 44, 46, 48, 51

[68] Hosomi, M., Yamagishi, H., Yamamoto, T., Bessho, K., Higo, Y., Yamane, K., Yamada, H., Shoji, M., Hachino, H., Fukumoto, C., Nagao, H., and Kano, H. A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram. In *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International* (Dec 2005), pp. 459–462. pages 26

[69] Hu, J., Xue, C., Zhuge, Q., Tseng, W.-C., and Sha, E. Data allocation optimization for hybrid scratch pad memory with sram and nonvolatile memory. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 21*, 6 (June 2013), 1094–1102. pages 45, 47, 56

[70] Hu, J., Xue, C., Zhuge, Q., Tseng, W.-C., and Sha, E.-M. Towards energy efficient hybrid on-chip scratch pad memory with non-volatile memory. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011* (March 2011), pp. 1–6. pages 9, 45, 55, 56

[71] Hu, J., Zhuge, Q., Xue, C., Tseng, W.-C., and Sha, E.-M. Optimizing data allocation and memory configuration for non-volatile memory based hybrid spm on embedded cmps. In *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International* (May 2012), pp. 982–989. pages 45, 47, 56

[72] Hu, J., Zhuge, Q., Xue, C., Tseng, W.-C., and Sha, E.-M. Optimizing data allocation and memory configuration for non-volatile memory based hybrid spm on embedded cmps. In *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International* (May 2012), pp. 982–989. pages 145

[73] Hu, J., Zhuge, Q., Xue, C. J., Tseng, W.-C., and Sha, E. H.-M. Management and optimization for nonvolatile memory-based hybrid scratchpad memory on multicore embedded processors. *ACM Trans. Embed. Comput. Syst. 13*, 4 (Mar. 2014), 79:1–79:25. pages 56

[74] Hubert, A., Nowak, E., Tachi, K., Maffini-Alvaro, V., Vizioz, C., Arvet, C., Colonna, J. P., Hartmann, J. M., Loup, V., Baud, L., Pauliac, S., Delaye, V., Carabasse, C., Molas, G., Ghibaudo, G., Salvo, B. D., Faynot, O., and Ernst, T. A stacked sonos technology, up to 4 levels and 6nm crystalline nanowires, with gate-all-around or independent gates (x03c6;-flash), suitable for full 3d integration. In *2009 IEEE International Electron Devices Meeting (IEDM)* (Dec 2009), pp. 1–4. pages 40

[75] Hubert, A., and SchÃĎfer, R. *Magnetic Domains: The Analysis of Magnetic Microstructures*, 1 ed. Springer-Verlag Berlin Heidelberg, 1998. pages 38

[76] Hung, Y.-C., Hsu, W.-T., Lin, T.-Y., and Fruk, L. Photoinduced write-once read-many-times memory device based on dna biopolymer nanocomposite. *Applied Physics Letters 99*, 25 (2011). pages 42

[77] Ilitzky, D. A., Hoffman, J. D., Chun, A., and Esparza, B. P. Architecture of the scalable communications core's network on chip. *IEEE Micro 27*, 5 (2007), 62–74. pages 93

[78] Imani, M., Patil, S., and Rosing, T. Low power data-aware stt-ram based hybrid cache architecture. In *2016 17th International Symposium on Quality Electronic Design (ISQED)* (March 2016), pp. 88–94. pages 118

[79] Ismail, M., Ahmed, E., Rana, A. M., Hussain, F., Talib, I., Nadeem, M. Y., Panda, D., and Shah, N. Improved endurance and resistive switching stability in ceria thin films due to charge transfer ability of al dopant. *ACS Applied Materials & Interfaces 8*, 9 (2016), 6127–6136. pages 124

[80] Jabeur, K., Buda-Prejbeanu, L. D., Prenat, G., and Pendina, G. D. Study of two writing schemes for a magnetic tunnel junction based on spin orbit torque. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering 7*, 8 (2013), 1054 – 1059. pages 29

[81] Jayapala, M., Barat, F., Aa, T., Catthoor, F., Corporaal, H., and Deconinck, G. Clustered loop buffer organization for low energy vliw embedded processors. *IEEE Transactions on Computers 54*, 6 (June 2005), 672–683. pages 65, 98

[82] Joy, A., Mishra, A. K., Xu, C., Xie, Y., Narayanan, V., Iyer, R., and Das, C. R. Cache revive: Architecting volatile stt-ram caches for enhanced performance in cmps. In *Proceedings of the 49th Annual Design Automation Conference* (New York, NY, USA, 2012), DAC '12, ACM, pp. 243–252. pages 47

[83] Kahng, A. B. The itrs design technology and system drivers roadmap: Process and status. In *Proceedings of the 50th Annual Design Automation Conference* (New York, NY, USA, 2013), DAC '13, ACM, pp. 34:1–34:6. pages vii, 3

[84] Kahng, K., and Sze, S. M. A floating gate and its application to memory devices. *IEEE Transactions on Electron Devices 14*, 9 (Sep 1967), 629–629. pages 22, 23

[85] Kamiya, K., Young Yang, M., Park, S.-G., Magyari-KÃűpe, B., Nishi, Y., Niwa, M., and Shiraishi, K. On-off switching mechanism of resistive random access memories based on the formation and disruption of oxygen vacancy conducting channels. *Applied Physics Letters 100*, 7 (2012). pages 32

[86] Kandemir, M., Ramanujam, J., Irwin, M. J., Vijaykrishnan, N., Kadayif, I., and Parikh, A. Dynamic management of scratch-pad memory space. In *Design Automation Conference, 2001. Proceedings* (2001), pp. 690–695. pages 53

[87] Kang, J. W., Lee, J. H., Lee, H. J., and Hwang, H. J. A study on carbon nanotube bridge as a electromechanical memory device. *Physica E: Low-dimensional Systems and Nanostructures 27*, 3 (2005), 332 – 340. pages 41

[88] Karandikar, A., and Parhi, K. K. Low power sram design using hierarchical divided bit-line approach. In *Proceedings. International Conference on Computer Design: VLSI in Computers and Processors, ICCD '98.* (October 1998), Intel Corp., pp. 82–88. pages 60

[89] Kato, Y., Yamada, T., and Shimada, Y. 0.18- mu;m nondestructive readout feram using charge compensation technique. *IEEE Transactions on Electron Devices 52*, 12 (Dec 2005), 2616–2621. pages 37

[90] Kawahara, T., Takemura, R., Miura, K., Hayakawa, J., Ikeda, S., Lee, Y., Sasaki, R., Goto, Y., Ito, K., Meguro, T., Matsukura, F., Takahashi, H., Matsuoka, H., and Ohno, H. 2mb spin-transfer torque ram (spram) with bit-by-bit bidirectional current write and parallelizing-direction current read. In *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers* (Feb 2007), pp. 480–617. pages 26

[91] Kawaharaa, T., Itoa, K., Takemuraa, R., and Ohno, H. Spin-transfer torque ram technology: Review and prospect. *Microelectronics Reliability: Advances in non-volatile memory technology 52*, 4 (April 2012), 613–627. pages 26

[92] Khvalkovskiy, A. V., Apalkov, D., Watts, S., Chepulskii, R., Beach, R. S., Ong, A., Tang, X., Driskill-Smith, A., Butler, W. H., Visscher, P. B., Lottis, D., Chen, E., Nikitin, V., and

Krounbi, M. Basic principles of STT-MRAM cell operation in memory arrays. *Journal of Physics D Applied Physics 46*, 7 (Feb. 2013), 074001. pages 26

[93] Khvalkovskiy, A. V., Apalkov, D., Watts, S., Chepulskii, R., Beach, R. S., Ong, A., Tang, X., Driskill-Smith, A., Butler, W. H., Visscher, P. B., Lottis, D., Chen, E., Nikitin, V., and Krounbi, M. Basic principles of STT-MRAM cell operation in memory arrays. *Journal of Physics D Applied Physics 46*, 7 (Feb. 2013), 074001. pages 146

[94] Kim, J., Chen, A., Behin-Aein, B., Kumar, S., Wang, J. P., and Kim, C. H. A technology-agnostic mtj spice model with user-defined dimensions for stt-mram scalability studies. In *Custom Integrated Circuits Conference (CICC), 2015 IEEE* (Sept 2015), pp. 1–4. pages 162

[95] Kim, J. H., Lim, W. C., Pi, U. H., Lee, J. M., Kim, W. K., Kim, J. H., Kim, K. W., Park, Y. S., Park, S. H., Kang, M. A., Kim, Y. H., Kim, W. J., Kim, S. Y., Park, J. H., Lee, S. C., Lee, Y. J., Yoon, J. M., Oh, S. C., Park, S. O., Jeong, S., Nam, S. W., Kang, H. K., and Jung, E. S. Verification on the extreme scalability of stt-mram without loss of thermal stability below 15 nm mtj cell. In *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers* (June 2014), pp. 1–2. pages 28

[96] Kim, K., and Lee, S. Integration of lead zirconium titanate thin films for high density ferroelectric random access memory. *Journal of Applied Physics 100*, 5 (2006). pages 37

[97] Kim, W., Jeong, J. H., Kim, Y., Lim, W. C., Kim, J. H., Park, J. H., Shin, H. J., Park, Y. S., Kim, K. S., Park, S. H., Lee, Y. J., Kim, K. W., Kwon, H. J., Park, H. L., Ahn, H. S., Oh, S. C., Lee, J. E., Park, S. O., Choi, S., Kang, H. K., and Chung, C. Extended scalability of perpendicular stt-mram towards sub-20nm mtj node. In *2011 International Electron Devices Meeting* (Dec 2011), pp. 24.1.1–24.1.4. pages 27

[98] Kim, W., Rao, S., et al. Manufacturable solution for high performance and highly scalable p-mtj mbit array paving the way for integration on advanced logic node n28 and beyond. In *Electron Devices Meeting, 2016. IEDM Technical Digest. IEEE International* (Dec to be published). pages 163

[99] Kin, J., Gupta, M., and Mangione-Smith, W. Filtering memory references to increase energy efficiency. *Computers, IEEE Transactions on 49*, 1 (Jan 2000), 1–15. pages 98

[100] Kin, J., Gupta, M., and Mangione-Smith, W. H. The filter cache: An energy efficient memory structure. In *Proceedings of the 30th Annual ACM/IEEE International Symposium on Microarchitecture* (Washington, DC, USA, 1997), MICRO 30, IEEE Computer Society, pp. 184–193. pages 54

[101] Kinoshita, K., Okutani, T., Tanaka, H., Hinoki, T., Agura, H., Yazawa, K., Ohmi, K., and Kishida, S. Flexible and transparent reram with gzo-memory-layer and gzo-electrodes on large pen sheet. In *Memory Workshop (IMW), 2010 IEEE International* (May 2010), pp. 1–3. pages 42

[102] Komalan, M., Pérez, J. I. G., Tenllado, C., Raghavan, P., Hartmann, M., and Catthoor, F. Feasibility exploration of nvm based i-cache through mshr enhancements. In *Proceedings of the Conference on Design, Automation & Test in Europe* (3001 Leuven, Belgium, 2014), DATE '14, pp. 21:1–21:6. pages xii, 144, 146, 154, 155

[103] Komalan, M., Sakhare, S., Bao, T. H., Rao, S., Kim, W., Tenllado, C., Gómez, J. I., Kar, G. S., Furnemont, A., and Catthoor, F. Cross-layer design and analysis of a low power, high density stt-mram for embedded systems. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (May 2017). (in press). pages 166

[104] Kroft, D. Lockup-free instruction fetch/prefetch cache organization. In *Proceedings of the 8th annual symposium on Computer Architecture* (Los Alamitos, CA, USA, 1981), ISCA '81, IEEE Computer Society Press, pp. 81–87. pages 126, 128

[105] Krounbi, M., Nikitin, V., Apalkov, D., Lee, J., Tang, X., Beach, R., Erickson, D., and Chen, E. (keynote) status and challenges in spin-transfer torque mram technology. *ECS Transactions 69*, 3 (2015), 119–126. pages 124

[106] Krounbi, M., Nikitin, V., Apalkov, D., Lee, J., Tang, X., Beach, R., Erickson, D., and Chen, E. Status and challenges in spin-transfer torque mram technology. *ECS transactions 69*, 3 (2015), 119–126. pages 11, 162

[107] Kultursay, E., Kandemir, M. T., Sivasubramaniam, A., and Mutlu, O. Evaluating stt-ram as an energy-efficient main memory alternative. In *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (2013), IEEE Computer Society, pp. 256–267. pages 10

[108] Kund, M., Beitel, G., Pinnow, C. U., Rohr, T., Schumann, J., Symanczyk, R., Ufert, K., and Muller, G. Conductive bridging ram (cbram): an emerging non-volatile memory technology scalable to sub 20nm. In *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.* (Dec 2005), pp. 754–757. pages 32

[109] Kwon, K.-W., Choday, S., Kim, Y., and Roy, K. Aware (asymmetric write architecture with redundant blocks): A high write speed stt-mram cache architecture. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 22*, 4 (April 2014), 712–720. pages 145

[110] Lam, C. H. The quest for the universal semiconductor memory. In *Electron Devices and Solid-State Circuits, 2005 IEEE Conference on* (Dec 2005), pp. 327–331. pages 6

[111] Lambrechts, A., Raghavan, P., Leroy, A., Talavera, G., Aa, T. V., Jayapala, M., Catthoor, F., Verkest, D., Deconinck, G., Corporaal, H., Robert, F., and Carrabina, J. Power breakdown analysis for a heterogeneous noc platform running a video application. In *Application-Specific Systems, Architecture Processors, 2005. ASAP 2005. 16th IEEE International Conference on* (July 2005), pp. 179–184. pages 2

[112] Lee, B. C., Ipek, E., Mutlu, O., and Burger, D. Architecting phase change memory as a scalable dram alternative. *SIGARCH Comput. Archit. News 37*, 3 (June 2009), 2–13. pages 10

[113] Lee, B. C., Zhou, P., Yang, J., Zhang, Y., Zhao, B., Ipek, E., Mutlu, O., and Burger, D. Phase-change technology and the future of main memory. *IEEE Micro 30*, 1 (Jan 2010), 143–143. pages 10

[114] Lee, H. Y., et al. Evidence and solution of over-reset problem for hfox based resistive memory with sub-ns switching speed and high endurance. In *IEEE International Electron Devices Meeting Technical Digest (IEDM), 2010* (December 2010), Tsing hua University, Taiwan, pp. 19.7.1 – 19.7.4. pages 107

[115] Lee, J.-S. Progress in non-volatile memory devices based on nanostructured materials and nanofabrication. *J. Mater. Chem. 21* (2011), 14097–14112. pages 30

[116] Li, J., Shi, L., Xue, C. J., Yang, C., and Xu, Y. Exploiting set-level write non-uniformity for energy-efficient nvm-based hybrid cache. In *2011 9th IEEE Symposium on Embedded Systems for Real-Time Multimedia* (Oct 2011), pp. 19–28. pages 119

[117] Li, L., Gao, L., and Xue, J. Memory coloring: a compiler approach for scratchpad memory management. In *14th International Conference on Parallel Architectures and Compilation Techniques (PACT'05)* (Sept 2005), pp. 329–338. pages 53

[118] Li, Q., Zhao, Y., Hu, J., Xue, C., Sha, E., and He, Y. Mgc: Multiple graph-coloring for non-volatile memory based hybrid scratchpad memory. In *Interaction between Compilers and Computer Architectures (INTERACT), 2012 16th Workshop on* (Feb 2012), pp. 17–24. pages 56

[119] Li, Y. T. S., Malik, S., and Wolfe, A. Cache modeling for real-time software: beyond direct mapped instruction caches. In *Real-Time Systems Symposium, 1996., 17th IEEE* (Dec 1996), pp. 254–263. pages 52

[120] Lin, C. W., Wang, D. Y., Tai, Y., Jiang, Y. T., Chen, M. C., Chen, C. C., Yang, Y. J., and Chen, Y. F. Type-II heterojunction organic/inorganic hybrid non-volatile memory based on FeS_2 nanocrystals embedded in poly(3-hexylthiophene). *Journal of Physics D Applied Physics 44* (July 2011), 292002. pages 41

[121] Lin, C.-Y., Wu, C.-Y., Wu, C.-Y., Tseng, T.-Y., and Hu, C. Modified resistive switching behavior of zro2 memory films based on the interface layer formed by using ti top electrode. *Journal of Applied Physics 102*, 9 (2007). pages 33

[122] Lin, H. T., Pei, Z., Chen, J. R., Hwang, G. W., Fan, J. F., and Chan, Y. J. A new nonvolatile bistable polymer-nanoparticle memory device. *IEEE Electron Device Letters 28*, 11 (Nov 2007), 951–953. pages 41

[123] Lin, H. T., Pei, Z., Chen, J. R., Kung, C. P., Lin, Y. C., Tseng, C. M., and Chan, Y. J. A 16-byte nonvolatile bistable polymer memory array on plastic substrates. In *Electron Devices Meeting, 2007. IEDM 2007. IEEE International* (Dec 2007), pp. 233–236. pages 42

[124] Lin, M. H., Wu, M. C., Lin, C. H., and Tseng, T. Y. Effects of vanadium doping on resistive switching characteristics and mechanisms of $snzro_3$ based memory films. *IEEE Transactions on Electron Devices 57*, 8 (Aug 2010), 1801–1808. pages 34

[125] Liu, L., Pai, C.-F., Li, Y., Tseng, H. W., Ralph, D. C., and Buhrman, R. A. Spin-torque switching with the giant spin hall effect of tantalum. *Science 336*, 6081 (2012), 555–558. pages 28

[126] Lodi, A., et al. Xisystem: a xirisc-based soc with reconfigurable io module. *IEEE Journal of Solid-State Circuits 41*, 1 (Jan 2006), 85–96. pages 93

[127] Lundqvist, T., and Stenström, P. An integrated path and timing analysis method based on cycle-level symbolic execution. *Real-Time Syst. 17*, 2-3 (Dec. 1999), 183–207. pages 52

[128] Lundqvist, T., and Stenstrom, P. Timing anomalies in dynamically scheduled microprocessors. In *Real-Time Systems Symposium, 1999. Proceedings. The 20th IEEE* (1999), pp. 12–21. pages 53

[129] Mangalagiri, P., Sarpatwari, K., Yanamandra, A., Narayanan, V., Xie, Y., Irwin, M. J., and Karim, O. A. A low-power phase change memory based hybrid cache architecture. In *Proceedings of the 18th ACM Great Lakes Symposium on VLSI* (2008), GLSVLSI '08, ACM, pp. 395–398. pages 9, 45, 55, 57

[130] Masselos, K., Catthoor, F., Goutis, C. E., and Deman, H. A systematic methodology for the application of data transfer and storage optimizing code transformations for power consumption and execution time reduction in realizations of multimedia algorithms on programmable processors. *IEEE Trans. Very Large Scale Integr. Syst. 10*, 4 (Aug. 2002), 515–518. pages 2

[131] Meena, J. S., Sze, S. M., Chand, U., and Tseng, T.-Y. Overview of emerging nonvolatile memory technologies. *Nanoscale Research Letters 9*, 1 (2014), 1–33. pages vii, xv, 2, 7, 9, 43, 44, 46, 48, 51

[132] Mei, B., Vernalde, S., Verkest, D., Man, H. D., and Lauwereins, R. Adres: An architecture with tightly coupled vliw processor and coarse-grained reconfigurable matrix. In *Proc. of Field-Programmable Logic and Applications* (2003), IMEC, pp. 61–70. pages 94

[133] Meier, G., Bolte, M., Eiselt, R., Krüger, B., Kim, D.-H., and Fischer, P. Direct imaging of stochastic domain-wall motion driven by nanosecond current pulses. *Phys. Rev. Lett. 98* (May 2007), 187202. pages 39

[134] Meza, J., Luo, Y., Khan, S., Zhao, J., Xie, Y., and Mutlu, O. A case for efficient hardware/software cooperative management of storage and memory. In *The Fifth Workshop on Energy Efficient Design* (Tel Aviv, Israel, 2013), WEED '13. pages 49, 50

[135] Min, K. K., Kwon, I. W., Cho, S., Kwon, M., Jang, T. S., Oh, T. K., Kim, Y. T., Cha, S. Y., Park, S. K., and Hong, S. J. Study on the sub-threshold margin characteristics of the extremely scaled 3-d dram cell transistors. In *Memory Workshop (IMW), 2015 IEEE International* (May 2015), pp. 1–4. pages 5

[136] Miron, I. M., Garello, K., Gaudin, G., Zermatten, P.-J., Costache, M. V., Auffret, S., Bandiera, S., Rodmacq, B., Schuhl, A., and Gambardella, P. Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection. *Nature 476*, 7359 (May 2011), 189–193. pages 28

[137] Miron, I. M., Gaudin, G., Auffret, S., Rodmacq, B., Schuhl, A., Pizzini, S., Vogel, J., and Gambardella, P. Current-driven spin torque induced by the rashba effect in a ferromagnetic metal layer. *Nature Materials 09*, 3 (March 2010), 230–234. pages 28

[138] Mittal, S., and Vetter, J. S. Ayush: A technique for extending lifetime of sram-nvm hybrid caches. *IEEE Computer Architecture Letters 14*, 2 (July 2015), 115–118. pages 47

[139] Moise, T. S., Summerfelt, S. R., McAdams, H., Aggarwal, S., Udayakumar, K. R., Celii, F. G., Martin, J. S., Xing, G., Hall, L., Taylor, K. J., Hurd, T., Rodriguez, J., Remack, K., Khan, M. D., Boku, K., Stacey, G., Yao, M., Albrecht, M. G., Zielinski, E., Thakre, M., Kuchimanchi, S., Thomas, A., McKee, B., Rickes, J., Wang, A., Grace, J., Fong, J., Lee, D., Pietrzyk, C., Lanham, R., Gilbert, S. R., Taylor, D., Amano, J., Bailey, R., Chu, F., Fox, G., Sun, S., and Davenport, T. Demonstration of a 4 mb, high density ferroelectric memory embedded within a 130 nm, 5 lm cu/fsg logic process. In *Electron Devices Meeting, 2002. IEDM '02. International* (Dec 2002), pp. 535–538. pages 37

[140] Monazzah, A., Farbeh, H., Miremadi, S., Fazeli, M., and Asadi, H. Ftspm: A fault-tolerant scratchpad memory. In *Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on* (2013), pp. 1–10. pages 47, 55, 56

[141] Mueller, F. Timing analysis for instruction caches. *Real-Time Systems 18*, 2 (2000), 217–247. pages 52

[142] Muralimanohar, N., and Balasubramonian, R. Cacti 6.0: A tool to model large caches, 2009. pages 61, 72

[143] Naji, P. K., Durlam, M., Tehrani, S., Calder, J., and DeHerrera, M. F. A 256 kb 3.0 v 1t1mtj nonvolatile magnetoresistive

ram. In *Solid-State Circuits Conference, 2001. Digest of Technical Papers. ISSCC. 2001 IEEE International* (Feb 2001), pp. 122–123. pages 25

[144] Nil, M. D., Yseboodt, L., Bouwens, F., Hulzink, J., Berekovic, M., Huisken, J., and van Meerbergen, J. Ultra low power asip design for wireless sensor nodes. In *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on* (Dec 2007), pp. 1352–1355. pages 2

[145] Nishimatsu, T., Waghmare, U. V., Kawazoe, Y., and Vanderbilt, D. Fast molecular-dynamics simulation for ferroelectric thin-film capacitors using a first-principles effective hamiltonian. *Phys. Rev. B 78* (Sep 2008), 104104. pages viii, 36

[146] Noguchi, H., Ikegami, K., Kushida, K., Abe, K., Itai, S., Takaya, S., Shimomura, N., Ito, J., Kawasumi, A., Hara, H., and Fujita, S. A 3.3ns-access-time 71.2w/mhz 1mb embedded stt-mram using physically eliminated read-disturb scheme and normally-off memory architecture. In *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers* (Feb 2015), pp. 1–3. pages 162

[147] Noguchi, H., Ikegami, K., Shimomura, N., Tetsufumi, T., Ito, J., and Fujita, S. Highly reliable and low-power nonvolatile cache memory with advanced perpendicular stt-mram for high-performance cpu. In *VLSI Circuits Digest of Technical Papers, 2014 Symposium on* (June 2014), pp. 1–2. pages 47, 144, 146

[148] Noguchi, H., Ikegami, K., Takaya, S., Arima, E., Kushida, K., Kawasumi, A., Hara, H., Abe, K., Shimomura, N., Ito, J., Fujita, S., Nakada, T., and Nakamura, H. 4mb stt-mram-based cache with memory-access-aware power optimization and write-verify-write / read-modify-write scheme. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)* (Jan 2016), pp. 132–133. pages 162

[149] Nowak, E. J., Aller, I., Ludwig, T., Kim, K., Joshi, R. V., Chuang, C.-T., Bernstein, K., and Puri, R. Turning silicon on its edge [double gate cmos/finfet technology]. *IEEE Circuits and Devices Magazine 20*, 1 (Jan 2004), 20–31. pages 4

[150] Nowak, J. J., Robertazzi, R. P., Sun, J. Z., Hu, G., Park, J. H., Lee, J., Annunziata, A. J., Lauer, G. P., Kothandaraman, R., OâĂŹSullivan, E. J., Trouilloud, P. L., Kim, Y., and Worledge, D. C. Dependence of voltage and size on write error rates in spin-transfer torque magnetic random-access memory. *IEEE Magnetics Letters 7* (2016), 1–4. pages 28

[151] Nowak, J. J., Robertazzi, R. P., Sun, J. Z., Hu, G., Park, J. H., Lee, J., Annunziata, A. J., Lauer, G. P., Kothandaraman, R., OÂĂŹSullivan, E. J., Trouilloud, P. L., Kim, Y., and Worledge, D. C. Dependence of voltage and size on write error rates in spin-transfer torque magnetic random-access memory. *IEEE Magnetics Letters 7* (2016), 1–4. pages 125, 144

[152] Oboril, F., Bishnoi, R., Ebrahimi, M., and Tahoori, M. B. Evaluation of hybrid memory technologies using sot-mram for on-chip cache hierarchy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34*, 3 (March 2015), 367–380. pages vii, 29

[153] others, P. K. . Exascale computing study: Technology challenges in achieving exascale systems. Tech. rep., DARPA Information Processing Techniques Office, Sep 2008. pages 9

[154] Ou, E. Emerging non-volatile memory technologies for reconfigurable architectures. In *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on* (August 2011), Univ. of Sydney, Sydney, NSW, Australia, pp. 1–4. pages 95

[155] Panagopoulos, G., Augustine, C., and Roy, K. Modeling of dielectric breakdown-induced time-dependent stt-mram performance degradation. In *69th Device Research Conference* (June 2011), pp. 125–126. pages 28

[156] Park, C., Kan, J. J., Ching, C., Ahn, J., Xue, L., Wang, R., Kontos, A., Liang, S., Bangar, M., Chen, H., Hassan, S., Gottwald, M., Zhu, X., Pakala, M., and Kang, S. H. Systematic optimization of 1 gbit perpendicular magnetic tunnel junction arrays for 28 nm embedded stt-mram and beyond. In *2015 IEEE International Electron Devices Meeting (IEDM)* (Dec 2015), pp. 26.2.1–26.2.4. pages 162

[157] Park, S. K. Technology scaling challenge and future prospects of dram and nand flash memory. In *Memory Workshop (IMW), 2015 IEEE International* (May 2015), pp. 1–4. pages 5

[158] Parkin, S., 2004-2007. pages 38

[159] Parkin, S., Jiang, X., Kaiser, C., Panchula, A., Roche, K., and Samant, M. Magnetically engineered spintronic sensors and memory. *Proceedings of the IEEE 91*, 5 (May 2003), 661–680. pages 38

[160] Parkin, S. S. P., Hayashi, M., and Thomas, L. Magnetic domain-wall racetrack memory. *Science 320*, 5873 (2008), 190–194. pages viii, 38, 39, 40

[161] Phadke, S., and Narayanasamy, S. Mlp aware heterogeneous memory system. In *2011 Design, Automation Test in Europe* (March 2011), pp. 1–6. pages 45

[162] Piccolboni, G., Molas, G., Garbin, D., Vianello, E., Cueto, O., Cagli, C., Traore, B., Salvo, B. D., Ghibaudo, G., and Perniola, L. Investigation of cycle-to-cycle variability in hfo2-based oxram. *IEEE Electron Device Letters 37*, 6 (June 2016), 721–723. pages 58

[163] Pirovano, A., Lacaita, A. L., Benvenuti, A., Pellizzer, F., Hudgens, S., and Bez, R. Scaling analysis of phase-change memory technology. In *Electron Devices Meeting, 2003. IEDM '03 Technical Digest. IEEE International* (Dec 2003), pp. 29.6.1–29.6.4. pages 34

[164] Pouchet, L.-N. Polybench: The polyhedral benchmark suite, 2012. pages 147

[165] Puaut, I., and Pais, C. Scratchpad memories vs locked caches in hard real-time systems: a qualitative and quantitative comparison. Research Report PI 1818, 2006. pages 53

[166] Puaut, I., and Pais, C. Scratchpad memories vs locked caches in hard real-time systems: a quantitative comparison. In *2007 Design, Automation Test in Europe Conference Exhibition* (April 2007), pp. 1–6. pages 52

[167] Puschner, P., and Burns, A. Guest editorial: A review of worst-case execution-time analysis. *Real-Time Systems 18*, 2 (2000), 115–128. pages 52

[168] Qureshi, M., Franceschini, M., Jagmohan, A., and Lastras, L. Preset: Improving performance of phase change memories by exploiting asymmetry in write times. In *Computer Architecture (ISCA), 2012 39th Annual International Symposium on* (june 2012), pp. 380–391. pages 95

[169] Qureshi, M. K. Pay-as-you-go: low-overhead hard-error correction for phase change memories. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture* (2011), pp. 318–328. pages 95

[170] Qureshi, M. K., Karidis, J., Franceschini, M., Srinivasan, V., Lastras, L., and Abali, B. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture* (New York, NY, USA, 2009), MICRO 42, ACM, pp. 14–23. pages 47

[171] Qureshi, M. K., Srinivasan, V., and Rivers, J. A. Scalable high performance main memory system using phase-change memory technology. In *Proceedings of the 36th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2009), ISCA '09, ACM, pp. 24–33. pages 95

[172] Raghavan, P., Lambrechts, A., Jayapala, M., Catthoor, F., and Verkest, D. Distributed loop controller for multithreading in unithreaded ilp architectures. *IEEE Transactions on Computers 58*, 3 (March 2009), 311–321. pages 65

[173] Raghavan, P., Lambrechts, A., Jayapala, M., Catthoor, F., Verkest, D., and Corporaal, H. Very wide register: An asymmetric register file organization for low power embedded processors. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2007* (April 2007), IMEC, pp. 1–6. pages 54, 67, 68, 107, 144, 147, 161

[174] Raoux, S., Burr, G. W., Breitwisch, M. J., Rettner, C. T., Chen, Y. C., Shelby, R. M., Salinga, M., Krebs, D., Chen, S. H., Lung, H. L., and Lam, C. H. Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development 52*, 4.5 (July 2008), 465–479. pages 34

[175] Raoux, S., Salinga, M., Jordan-Sweet, J. L., and Kellock, A. Effect of al and cu doping on the crystallization properties of the phase change materials sbte and gesb. *Journal of Applied Physics 101*, 4 (2007). pages 34

[176] Raychowdhury, A. Pulsed read in spin transfer torque (stt) memory bitcell for lower read disturb. In *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)* (July 2013), pp. 34–35. pages 28

[177] Roy, A. S., Sarkar, A., and Mudanai, S. P. Compact modeling of magnetic tunneling junctions. *IEEE Transactions on Electron Devices 63*, 2 (Feb 2016), 652–658. pages 162

[178] Roy, A. S., Sarkar, A., and Mudanai, S. P. Compact modeling of magnetic tunneling junctions. *IEEE Transactions on Electron Devices 63*, 2 (Feb 2016), 652–658. pages 162

[179] Samavatian, M. H., Abbasitabar, H., Arjomand, M., and Sarbazi-Azad, H. An efficient stt-ram last level cache architecture for gpus. In *Proceedings of the 51st Annual Design Automation Conference* (2014), DAC '14, ACM, pp. 197:1–197:6. pages 145

[180] Santini, C. A., Sebastian, A., Marchiori, C., Jonnalagadda, V. P., Dellmann, L., Koelmans, W. W., Rossell, M. D., Rossel, C. P., and Eleftheriou, E. Oxygenated amorphous carbon for resistive memory applications. *Nature Communications 6* (Oct 2015). pages 33

[181] Sarpeshkar, R., Delbruck, T., and Mead, C. A. White noise in mos transistors and resistors. *IEEE Circuits and Devices Magazine 9*, 6 (November 1993), 23–29. pages 59

[182] Sawa, A. Resistive switching in transition metal oxides. *Materials Today 11*, 6 (2008), 28 – 36. pages 33

[183] Seo, S., Lee, M. J., Seo, D. H., Jeoung, E. J., Suh, D.-S., Joung, Y. S., Yoo, I. K., Hwang, I. R., Kim, S. H., Byun, I. S., Kim, J.-S., Choi, J. S., and Park, B. H. Reproducible resistance switching in polycrystalline nio films. *Applied Physics Letters 85*, 23 (2004), 5655–5657. pages 34

[184] Seong, N. H., Woo, D. H., and Lee, H.-H. S. Security refresh: Prevent malicious wear-out and increase durability for phase-change memory with dynamically randomized address mapping. In *Proceedings of the 37th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2010), ISCA '10, ACM, pp. 383–394. pages 47

[185] Setter, N., Damjanovic, D., Eng, L., Fox, G., Gevorgian, S., Hong, S., Kingon, A., Kohlstedt, H., Park, N. Y., Stephenson, G. B., Stolitchnov, I., Taganstev, A. K., Taylor, D. V., Yamada, T., and Streiffer, S. Ferroelectric thin films: Review of materials, properties, and applications. *Journal of Applied Physics 100*, 5 (2006). pages 37

[186] Setter, N., et al. Ferroelectric thin films: Review of materials, properties, and applications. *Journal of Applied Physics 100*, 5 (January 2006), 051604. pages 9

[187] Shahidi, G. Design-technology interaction for post-32 nm node cmos technologies. In *VLSI Technology (VLSIT), 2010 Symposium on* (2010), IBM, pp. 143–144. pages xi, 120, 121

[188] Sheikholeslami, A., and Gulak, P. G. A survey of circuit innovations in ferroelectric random-access memories. *Proceedings of the IEEE 88*, 5 (May 2000), 667–689. pages 36

[189] Sheu, S.-S., Chang, M.-F., Lin, K.-F., Wu, C.-W., Chen, Y.-S., Chiu, P.-F., Kuo, C.-C., Yang, Y.-S., Chiang, P.-C., Lin, W.-P., Lin, C.-H., Lee, H.-Y., Gu, P.-Y., Wang, S.-M., Chen, F., Su, K.-L., Lien, C.-H., Cheng, K.-H., Wu, H.-T., Ku, T.-K., Kao, M.-J., and Tsai, M.-J. A 4mb embedded slc resistive-ram macro with 7.2ns read-write random-access time and 160ns mlc-access capability. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011* (February 2011), ITRI, pp. 200–202. pages 122

[190] Sheu, S.-S., Cheng, K.-H., Chang, M.-F., Chiang, P.-C., Lin, W.-P., Lee, H.-Y., Chen, P.-S., Chen, Y.-S., Wu, T.-Y., Chen, F., Su, K.-L., Kao, M.-J., and Tsai, M.-J. Fast-write resistive ram (rram) for embedded applications. *IEEE Design and Test of Computers 28*, 1 (February 2011), 64–71. pages 9, 75, 107

[191] Shipway, A. N., Katz, E., and Willner, I. *Molecular Machines and Motors*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, ch. Molecular Memory and Processing Devices in Solution and on Surfaces, pp. 237–281. pages 41

[192] Shulaker, M. M., Wu, T. F., Pal, A., Zhao, L., Nishi, Y., Saraswat, K., Wong, H. S. P., and Mitra, S. Monolithic 3d integration of logic and memory: Carbon nanotube fets, resistive ram, and silicon fets. In *2014 IEEE International Electron Devices Meeting* (Dec 2014), pp. 27.4.1–27.4.4. pages 43

[193] Shulaker, M. M., Wu, T. F., Sabry, M. M., Wei, H., Wong, H. S. P., and Mitra, S. Monolithic 3d integration: A path from concept to reality. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2015* (March 2015), pp. 1197–1202. pages 9, 43

[194] Singh, H., et al. Morphosys: an integrated reconfigurable system for data-parallel and computation-intensive applications. *Computers, IEEE Transactions on 49*, 5 (May 2000), 465–481. pages 93

[195] Smullen, C. W., , Mohan, V., Nigam, A., Gurumurthi, S., and Stan, M. R. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *IEEE 17th International Symposium on High Performance Computer Architecture (HPCA), 2011* (February 2011), Univ. of Virginia, pp. 50–61. pages 9, 55

[196] Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. The missing memristor found. *Nature 453*, 7191 (May 2008), 80–83. pages 29

[197] Sun, G., Dong, X., Xie, Y., Li, J., and Chen, Y. A novel architecture of the 3d stacked mram l2 cache for cmps. In *2009 IEEE 15th International Symposium on High Performance Computer Architecture* (Feb 2009), pp. 239–249. pages 28

[198] Sun, H., Liu, C., Xu, W., Zhao, J., Zheng, N., and Zhang, T. Using magnetic ram to build low-power and soft error-resilient l1 cache. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 20*, 1 (Jan 2012), 19–28. pages 45, 55, 57, 146

[199] Tang, W., Shi, H., Xu, G., Ong, B., Popovic, Z., Deng, J., Zhao, J., and Rao, G. Memory effect and negative differential resistance by electrode- induced two-dimensional single- electron tunneling in molecular and organic electronic devices. *Advanced Materials 17*, 19 (2005), 2307–2311. pages 41

[200] Technologies, E. Everspin technologies mram 4mb, 2009. pages 25

[201] technologies, T. Target base core processor manual. Tech. rep., TARGET, Leuven, Belgium, 2010. pages 63

[202] Tehrani, S., Engel, B., Slaughter, J. M., Chen, E., DeHerrera, M., Durlam, M., Naji, P., Whig, R., Janesky, J., and Calder, J. Recent developments in magnetic tunnel junction mram. *IEEE Transactions on Magnetics 36*, 5 (Sep 2000), 2752–2757. pages 25

[203] Texas Instruments. *TMS320C64x/C64x+ DSP CPU and Instruction Set : Reference guide*, July 2010. pages 146

[204] Udayakumaran, S., and Barua, R. Compiler-decided dynamic memory allocation for scratch-pad based embedded systems. In *Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems* (New York, NY, USA, 2003), CASES '03, ACM, pp. 276–286. pages 53

[205] Uh, G.-R., Wang, Y., Whalley, D., Jinturkar, S., Burns, C., and Cao, V. Effective exploitation of a zero overhead loop buffer. In *Workshop on Languages, Compilers, and Tools for Embedded Systems* (1999), Lucent Technologies, pp. 10–19. pages 54, 65, 98

[206] Venkatesan, R., Sharad, M., Roy, K., and Raghunathan, A. Dwm-tapestri - an energy efficient all-spin cache using domain wall shift

based writes. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2013), pp. 1825–1830. pages 39

[207] Verma, M., and Marwedel, P. *Advanced Memory Optimization Techniques for Low-Power Embedded Processors*, vol. 1. Springer Netherlands, 2007. pages 2

[208] Vettiger, P., Cross, G., Despont, M., Drechsler, U., Durig, U., Gotsmann, B., Haberle, W., Lantz, M. A., Rothuizen, H. E., Stutz, R., and Binnig, G. K. The "millipede" - nanotechnology entering data storage. *IEEE Transactions on Nanotechnology 1*, 1 (Mar 2002), 39–55. pages 42

[209] Wang, F., and Wu, X. Non-volatile memory devices based on chalcogenide materials. In *2009 Sixth International Conference on Information Technology: New Generations* (April 2009), pp. 5–9. pages viii, 35

[210] Wang, J., Tim, Y., Wong, W. F., Ong, Z. L., Sun, Z., and Li, H. H. A coherent hybrid sram and stt-ram l1 cache architecture for shared memory multicores. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)* (Jan 2014), pp. 610–615. pages 145

[211] Wang, P., Sun, G., Wang, T., Xie, Y., and Cong, J. Designing scratchpad memory architecture with emerging stt-ram memory technologies. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on* (2013), pp. 1244–1247. pages 47, 55, 57

[212] WANG, S.-Y., and TSENG, T.-Y. Interface engineering in resistive switching memories. *Journal of Advanced Dielectrics 01*, 02 (2011), 141–162. pages 33

[213] Wang, Z., Gu, Z., Yao, M., and Shao, Z. Endurance-aware allocation of data variables on nvm-based scratchpad memory in real-time embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34*, 10 (Oct 2015), 1600–1612. pages 45

[214] Wikipedia. Magnetoresistive random-access memory, 2007. pages vii, 25

[215] Wong, H.-S. P., and Salahuddin, S. Memory leads the way to better computing. *Nature Nanotechnology 10*, 3 (July 2015), 191–194. pages vii, 9, 31

[216] Wu, M. C., Lin, Y. W., Jang, W. Y., Lin, C. H., and Tseng, T. Y. Low-power and highly reliable multilevel operation in zro_2 1t1r rram. *IEEE Electron Device Letters 32*, 8 (Aug 2011), 1026–1028. pages 34

[217] Xu, C., Dong, X., Jouppi, N. P., and Xie, Y. Design implications of memristor-based rram cross-point structures. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011* (March 2011), pp. 1–6. pages 29, 34

[218] Xu, N., Wang, J., Lu, Y., Park, H. H., Fu, B., Chen, R., Choi, W., Apalkov, D., Lee, S., Ahn, S., Kim, Y., Nishizawa, Y., Lee, K. H., Park, Y., and Jung, E. S. Physics-based compact modeling framework for state-of-the-art and emerging stt-mram technology. In *2015 IEEE International Electron Devices Meeting (IEDM)* (Dec 2015), pp. 28.5.1–28.5.4. pages 162

[219] Xu, Z., Yang, C., Mao, M., Sutaria, K. B., Chakrabarti, C., and Cao, Y. Compact modeling of stt-mtj devices. *Solid-State Electronics 102* (2014), 76 – 81. Selected papers from {ESSDERC} 2013. pages 162

[220] Yoon, H. Row buffer locality aware caching policies for hybrid memories. In *Proceedings of the 2012 IEEE 30th International Conference on Computer Design (ICCD 2012)* (Washington, DC, USA, 2012), ICCD '12, IEEE Computer Society, pp. 337–344. pages viii, 47, 48

[221] Yoshida, C., Kurasawa, M., Lee, Y. M., Tsunoda, K., Aoki, M., and Sugiyama, Y. A study of dielectric breakdown mechanism in cofeb/mgo/cofeb magnetic tunnel junction. In *2009 IEEE International Reliability Physics Symposium* (April 2009), pp. 139–142. pages 28

[222] Yu, S., and Chen, P. Y. Emerging memory technologies: Recent trends and prospects. *IEEE Solid-State Circuits Magazine 8*, 2 (Spring 2016), 43–56. pages xv, 43, 44, 46, 48, 51

[223] Yu, Z., Li, W., Hagen, J. A., Zhou, Y., Klotzkin, D., Grote, J. G., and Steckl, A. J. Photoluminescence and lasing from deoxyribonucleic acid (dna) thin films doped with sulforhodamine. *Appl. Opt. 46*, 9 (Mar 2007), 1507–1513. pages 41

[224] Yun, J.-G., Lee, J. D., and Park, B.-G. 3d {NAND} flash memory with laterally-recessed channel (lrc) and connection gate architecture. *Solid-State Electronics 55*, 1 (2011), 37 – 43. pages 40

[225] Zhao, J., Xu, C., and Xie, Y. Bandwidth-aware reconfigurable cache design with hybrid memory technologies. In *Computer-Aided Design*

(ICCAD), 2011 IEEE/ACM International Conference on (2011), pp. 48–55. pages 95

[226] Zhu, G., Lu, K., and Li, X. *SCM-BSIM: A Non-Volatile Memory Simulator Based on BOCHS*. Springer New York, New York, NY, 2013, pp. 977–984. pages 162

ESAT-INSYS (KU LEUVEN), DACYA (UC MADRID)
komalan.manuperumkunnil@esat.kuleuven.be, komalanm@ucm.es